ARMY RESEARCH LABORATORY

A Tutorial on Neural Networks Using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) Training Algorithm and Molecular Descriptors With Application to the Prediction of Dielectric Constants Through the Development of Quantitative Structure Property Relationships (QSPRs)

by Robert C. Schweitzer
and Jeffrey B. Morris

ARL-TR-2155 January 2000

20000727 296

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 4

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Abstract

The use of quantitative structure property relationships (QSPRs) is proposed for the calculation of dielectric constants. A data set of 497 compounds with a wide variety of functional groups is assembled. These compounds span the dielectric constant range of 1–40. A total of 65 molecular descriptors is calculated for these compounds. These descriptors include the dipole moment, polarizability, counts of elemental types, an indicator of hydrogen bonding capability, charged partial surface area descriptors, and molecular connectivity descriptors. Subsets of these descriptors are used to build models in an attempt to find the best possible correlation between chemical structure and dielectric constant. A total of 70,000 models is examined. Neural networks using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) training algorithm are employed to build the models. A total of 191 models has test set errors less than 2.0 and training set errors less than 3.0, where the errors are calculated as the mean of the absolute values of the residuals for sets of 97 and 350 compounds, respectively.

Acknowledgments

The authors would like to thank the Department of Defense (DOD) Major Shared Resource Center (MSRC) at the U.S. Army Research Laboratory (ARL) for use of the Silicon Graphics Incorporated (SGI) Origin 2000 cluster. Dr. Gary Small of the Center for Intelligent Chemical Instrumentation at Ohio University is acknowledged for providing the database of three-dimensional (3-D) coordinates and the accompanying software. Parts of this work were presented at the 1998 Pittsburgh Conference and Exposition on Analytical Chemistry and Applied Spectroscopy, New Orleans, LA, 1998. Robert Schweitzer acknowledges support through the National Academy of Sciences/National Research Council Postdoctoral Associateship Program at ARL.

INTENTIONALLY LEFT BLANK.

Table of Contents

INTENTIONALLY LEFT BLANK.

List of Figures

INTENTIONALLY LEFT BLANK.

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

The dielectric constant, or relative permittivity, is an important fundamental molecular property that can be a useful predictor of the behavior of substances on a macroscopic scale. The primary importance of the dielectric constant lies in the fact that it is a measure of the ability of a substance to maintain a charge separation. Dielectric constants can be measured experimentally through the use of a capacitance cell according to the following equation:

$$C = \varepsilon C_o , \tag{1}$$

where C_o is the capacitance of the cell under vacuum, C is the capacitance of the cell when filled with the dielectric medium, and ε is the dielectric constant. Since the charge on the plates of the capacitance cell does not change, the net effect of the presence of the dielectric material is to reduce the effective electric field across the plates:

$$E = E_o / \varepsilon , \tag{2}$$

where E_o is the electric field strength under vacuum and E is the net electric field strength when the dielectric medium is present.

The dielectric constant is particularly important for the interpretation of certain solvent-solute behavior. Solvents can be divided into two general classes: polar and nonpolar. Solvents with dielectric constants less than about 15 are considered to be nonpolar [1]. The reduction of net electric field strength within high dielectric media (see equation [2]) also applies to point charges. It is well known that the high solubilities exhibited by ionic salts in polar solvents are due to reduction of the electrostatic forces between oppositely charged salt ions. The large dielectric constant of a polar solvent effectively shields the ions from each other. The lower dielectric constants of nonpolar solvents do not provide sufficient reduction in the effective electrostatic forces to stabilize the separated ions in solution. In this case, the ions remain highly associated as low-solubility solid-phase salts. Solvent polarity is also known to play a role in the processing of ionomers, particularly with regard to melt viscosity behavior [2]. The solvation properties of supercritical water have received much attention over the past several decades. Measurements of the dielectric constant of water over a wide range of temperatures and pressures indicate that water behaves as a polar solvent ($\varepsilon > 80$) at ambient conditions and also as a nonpolar solvent ($\varepsilon < 10$) in its supercritical state [3, 4]. Salts that are soluble in water at ambient conditions become highly associated and precipitate out of solution under supercritical conditions [5]. It is also believed that the solvent dielectric constant plays a significant, although less predictable, role in the solubility of many highly polar covalent compounds, especially those with large variations in surface charge distribution.

Unfortunately, dielectric constant values are not always readily available in the literature. Early calculational efforts were based on Debye's dielectric theory. Expressions such as the Clausius-Mosotti equation are generally only useful for dilute gases and some liquids of limited polarity. A significant improvement is made through the use of the Onsager equation:

$$\frac{\left(\varepsilon - n^2\right)\left(2\varepsilon + n^2\right)}{\varepsilon\left(n^2 + 2\right)^2} = \frac{4\pi N\mu^2}{9kT}, \tag{3}$$

where n is the refractive index, N is the number of atoms in a unit volume, μ is the dipole moment, k is Boltzmann's constant, and T is temperature. While the Onsager equation works reasonably well, even for some polar liquids, it does not sufficiently take into account intermolecular interactions and is especially poor with strong hydrogen-bonding liquids, such as water or alcohols. The limitations of equation (3) have been verified by calculating dielectric constants for a group of hydrocarbons and a group of compounds that exert strong intermolecular attractions. The calculated values for the group of hydrocarbons agree very well with experimental values from the literature, but the calculated values for the second group of compounds display very poor agreement with the literature. Computational efforts involving computer simulation and molecular dynamics are also limited for hydrogen-bonding solvents, due to the sensitivity of the dielectric constant on long-range intermolecular interactions [6]. The current state of theoretical approaches simply does not allow their use as general purpose tools for calculating dielectric constants for a wide variety of compounds.

A solution to this problem is the use of quantitative structure property relationships (QSPRs). A QSPR is essentially a calibration model in which the independent variables are molecular descriptors that describe the structure of the molecules and the dependent variable is the property of interest. The development of a QSPR depends upon the availability of a set of compounds (the calibration set) for which the value of the property of interest for each is known and the necessary molecular descriptors can be calculated. Given a QSPR, property values can be predicted for compounds that are not present in the calibration set.

The roots of QSPR/quantitative structure activity relationships (QSARs) date back to the 1800s, but the key paper that instigated the current flurry of activity was published in 1962 by Hansch et al. [7]. In its broadest definition, a QSAR/QSPR is an attempt to correlate chemical structure with activity or property. An article by Tute [8] gives an excellent history and a good introduction to QSARs. The literature is filled with QSARs for biological activities related to medicine and environmental studies, as well as QSPRs related to a wide variety of physical and chemical properties. An area of much debate has been the way in which the chemical structure is encoded. This debate has resulted in two differing approaches to the field of QSAR. The Hansch method, which was the first approach, uses experimental parameters to represent the chemical structure [9]. These parameters encode the hydrophobic, electronic, and steric forces of a molecule and thus yield models that give an intuitive explanation of the effect of each parameter in the model. The most famous of these parameters is the octanol/water partition coefficient, which represents the hydrophobicity/hydrophilicity of a molecule. One disadvantage of this approach is that data must either be available or be obtained by experiments for every member of the calibration set. This disadvantage has been addressed by the development of programs that can calculate approximations to the experimental parameters [10]. Another major limitation of the Hansch approach is that the molecules used to build the model must have the same basic backbone structure with varying substituents (i.e., must be congeneric). An oft-cited advantage to the Hansch approach is the fact that, by always using the same small set of descriptors, the models that are generated can be easily compared to one another.

The second approach to the development of QSPRs allows a wide variety of descriptors that is directly calculated from the chemical structure. It is this approach that is used in this paper. One advantage to this approach is that, since experimental parameters are not needed, any molecule can be easily encoded. A disadvantage to this approach is that the descriptors used do not always give an intuitive physical description of the effect of each parameter in the model. In addition, the models generated are not easily comparable to one another because there is no standard set of descriptors that is always used to build the models. The advantage of allowing a wide variety of descriptors is that the set of descriptors used for a given model can be tailored to the model and, as a result, more accurate models can be established. A major focus of this paper is the search for a set of molecular descriptors that gives the best possible correlation between molecular structure and dielectric constant.

The development of a QSPR requires a mathematical technique to build the model. There are many classical multivariate calibration techniques available that could be used. These include multivariate linear regression, partial least-squares regression, and principal components regression [11–13]. The equation for multivariate linear regression is

$$Y = a_0 + a_1X_1 + a_2X_2 + \dots + a_nX_n, \tag{4}$$

where Y is the output or dependent variable, $X_1, \dots, X_n$ are the inputs or independent variables, and $a_0, \dots, a_n$ are regression coefficients calculated from a set of data (the calibration set) in which both the independent and dependent variables are known.

The model building technique used in this research, however, is the neural network. Whereas classical modeling requires a knowledge of the regression formula (such as the linear relationship in equation [4]), the neural network does not. The advantage is that the neural network develops a nonlinear relationship among the input variables without requiring a predefined relationship specified by the user. Neural networks are patterned after the connections of neurons in the brain and have an input layer of neurons, one or more intermediary layers of neurons (hidden layers), and an output layer of neurons. Each of the neurons in a given layer is connected to each of the neurons in the following layer, and each connection is assigned a weight. The weight values are to neural networks what regression coefficients are to a linear regression model. The process of determining the weights is an iterative process known as training the neural network, and this process is described in more detail later.

2. Experimental

One of the steps necessary for this research was the assembly of a data set. A list of 676 compounds was created for which condensed-phase static dielectric constant data were available from the *CRC Handbook of Chemistry and Physics* and the *Handbook of Organic Chemistry* [14, 15]. No limits were placed on the functional groups included, and, consequently, a very wide range of compounds resulted. The elements represented include hydrogen, carbon, nitrogen, oxygen, fluorine, chlorine, bromine, iodine, and sulfur. The *CRC Handbook of Chemistry and Physics* references the National Bureau of Standards (NBS) Circular No. 514 as its source of dielectric constant data [16]. The references from which this circular compiles its data range in publication date from 1892 to 1950. The NBS rates the probable accuracy of these dielectric constant data as ranging from better than 0.5% to worse than 2.0%, depending upon the compound. The temperatures at which the dielectric constants were measured range from −89° C to 220° C, with 615 of the compounds falling in the temperature range from 0° C to 40° C.

Some of the molecular descriptors used in this work require three-dimensional (3-D) coordinates. A large database of 3-D coordinates was supplied by the Center for Intelligent Chemical Instrumentation located at Ohio University [17]. The coordinates in this database were obtained by the use of a molecular mechanics package, MM2, developed by Allinger at the University of Georgia

[18]. Of the 676 compounds with dielectric constants, 565 have reasonable 3-D coordinates. Some of the molecular descriptors also require electronic data such as dipole moments, polarizabilities, and partial atomic charges. These values were calculated using the quantum mechanics package, *Gaussian 94 User's Reference Guide* [19]. Hartree-Fock self-consistent field calculations using the 6-31G basis set were successful for 540 of the compounds.

The computations described in this paper were implemented on Silicon Graphics Incorporated (SGI) computer systems located at the U.S. Army Research Laboratory (ARL) at Aberdeen Proving Ground (APG), MD. One of these systems is a Power Challenge Series with 12 R8000 processors running under Irix (version 6.2, SGI; Mountain View, CA). The other system consists of a cluster of five SGI Origin 2000s, with a total of 288 R10000 processors. The neural network software and the molecular descriptor software used in this work were developed using FORTRAN 77. The database software was supplied by the Center for Intelligent Chemical Instrumentation at Ohio University.

3. Results and Discussion

3.1 Molecular Descriptors. Molecular descriptors are often classified as topological, geometric, or electronic descriptors. A good overview of a variety of useful descriptors in each of these three classes is given by Katritzky and Gordeeva [20] in a study in which he examines the use of these descriptors to build models for five physicochemical properties and four biological activities. Jurs and coworkers have also published many articles [21–24] in which QSPRs are developed using calculated molecular descriptors. Topological descriptors include the count of the number of atoms of a particular elemental type in a given structure, the count of the number of occurrences of a particular bond type, the count of the number of occurrences of a particular structural fragment or functional group, molecular weight, molecular connectivity descriptors [25–28], and a variety of related descriptors [29–31]. Geometric descriptors include moments of inertia in which the principal axes of a structure are calculated [32], various calculations of molecular volume [32, 33], shadow indices [34], and total solvent accessible surface area [35].

Electronic descriptors are generally calculated using quantum mechanics and are developed from data such as atomic charges, highest occupied molecular orbital (HOMO), and lowest occupied molecular orbital (LOMO) energies, electron densities, superdelocalizabilities, polarizabilities, dipole moments, and various calculated energies. Karelson, Lobanov, and Katritzky [36] have published an article in which these calculations and descriptors are thoroughly reviewed. Some descriptors do not fit neatly into one of these three classes. Stanton and Jurs have developed a charged partial surface area (CPSA) descriptor, which combines geometric and electronic information [37]. Another mixed descriptor is the electronic-topological descriptor, which is patterned after the molecular connectivity descriptors, with the exception that atomic charges are used in the calculation rather than valences [38]. The descriptors used in this study are listed in Table 1 and include topological descriptors, electronic descriptors, and the CPSA descriptors of Stanton and Jurs. The grouping of the descriptors in this table, however, is by three artificial classes. The first class (descriptors 1–12) includes the dipole moment, polarizability, counts of elemental types, and the hydrogen bond descriptor. In an earlier study, these descriptors were the only descriptors used and have been termed the "simple" descriptors. Since then, software has been developed to allow the calculation of other types of descriptors such as CPSA descriptors (the second class, descriptors 13–39) and molecular connectivity descriptors (the third class, descriptors 40–65).

Some of the topological descriptors, such as counts of elemental types, need no explanation. The hydrogen bond descriptor is a binary descriptor for which a value of 1 indicates that a given structure has either a hydroxyl group or a primary or secondary amine group. The remaining topological descriptors used in this work are valence-corrected connectivity descriptors as developed by Kier and Hall [25, 26]. This class of descriptor is very widely used in QSPRs. The forerunner to this descriptor is the Wiener descriptor, which was developed in 1947 [39]. The Wiener descriptor is the summation of the number of bonds in the shortest path between two carbon atoms for all pairs of carbon atoms in the structure. In most current studies, the Wiener index is expanded to include all nonhydrogen atoms, not just carbon atoms. In Wiener's original study, this descriptor, along with a second, was found to correlate with boiling point for a set of alkanes. In 1975, Randić [40] developed a significantly new and useful topological descriptor, which he termed "the branching index." This descriptor, χ, which is now known as the molecular connectivity descriptor, is defined in the following equation:

Table 1. Set of 65 Molecular Descriptors

No.	Descriptor	Description
1	Dipole Moment	—
2	Polarizability	—
3	Carbon	Count of the number of carbons
4	Hydrogen	Count of the number of hydrogens
5	Oxygen	Count of the number of oxygens
6	Nitrogen	Count of the number of nitrogens
7	Fluorine	Count of the number of fluorines
8	Chlorine	Count of the number of chlorines
9	Bromine	Count of the number of bromines
10	Iodine	Count of the number of iodines
11	Sulfur	Count of the number of sulfurs
12	Hydrogen Bond	Presence of a hydroxyl group or a 1° or 2° amine
13	PPSA-1[a]	Sum of surface area for positively charged atoms
14	PNSA-1[a]	Sum of surface area for negatively charged atoms
15	PPSA-2[a]	Sum of surface area for positively charged atoms * sum of positive charges
16	PNSA-2[a]	Sum of surface area for negatively charged atoms * sum of negative charges
17	PPSA-3[a]	Sum of each (surface area for positively charged atom * positive charge)
18	PNSA-3[a]	Sum of each (surface area for negatively charged atom * negative charge)
19	DPSA-1[a]	PPSA-1 - PNSA-1
20	DPSA-2[a]	PPSA-2 - PNSA-2
21	DPSA-3[a]	PPSA-3 - PNSA-3
22	FPSA-1[a]	PPSA-1/total surface area
23	FNSA-1[a]	PNSA-1/total surface area
24	FPSA-2[a]	PPSA-2/total surface area
25	FNSA-2[a]	PNSA-2/total surface area
26	FPSA-3[a]	PPSA-3/total surface area
27	FNSA-3[a]	PNSA-3/total surface area
28	WPSA-1[a]	(PPSA-1 * total surface area)/1,000
29	WNSA-1[a]	(PNSA-1 * total surface area)/1,000
30	WPSA-2[a]	(PPSA-2 * total surface area)/1,000
31	WNSA-2[a]	(PNSA-2 * total surface area)/1,000
32	WPSA-3[a]	(PPSA-3 * total surface area)/1,000
33	WNSA-3[a]	(PNSA-3 * total surface area)/1,000
34	Rel Pos	Most positive charge/sum of positive charges
35	Rel Neg	Most negative charge/sum of negative charges
36	RPCS[a]	Surface area of most positively charged atom * most positive charge

[a] Symbols are taken from Stanton and Jurs [37].

Table 1. Set of 65 Molecular Descriptors (continued)

No.	Descriptor	Description
37	RNCS[a]	Surface area of most negatively charged atom * most negative charge
38	Sum Pos	Sum of all positive charges
39	Sum Neg	Sum of all negative charges
40	${}^{0}\chi^{v}$	Connectivity for atoms
41	${}^{1}\chi^{v}$	Connectivity for bonds (paths with two atoms)
42	${}^{2}\chi^{v}$	Connectivity for two bond paths (paths with three atoms)
43	${}^{3}\chi^{v}_{c}$	Connectivity for clusters with two bond paths
44	${}^{4}\chi^{v}_{s}$	Connectivity for stars with two bond paths
45	${}^{3}\chi^{v}_{r}$	Connectivity for three bond rings
46	${}^{3}\chi^{v}$	Connectivity for three bond paths
47	${}^{4}\chi^{v}_{c}$	Connectivity for clusters with three bond paths
48	${}^{5}\chi^{v}_{s}$	Connectivity for stars with three bond paths
49	${}^{4}\chi^{v}_{r}$	Connectivity for four bond rings
50	${}^{4}\chi^{v}$	Connectivity for four bond paths
51	${}^{5}\chi^{v}_{c}$	Connectivity for clusters with four bond paths
52	${}^{6}\chi^{v}_{s}$	Connectivity for stars with four bond paths
53	${}^{5}\chi^{v}_{r}$	Connectivity for five bond rings
54	${}^{5}\chi^{v}$	Connectivity for five bond paths
55	${}^{6}\chi^{v}_{c}$	Connectivity for clusters with five bond paths
56	${}^{7}\chi^{v}_{s}$	Connectivity for stars with five bond paths
57	${}^{6}\chi^{v}_{r}$	Connectivity for six bond rings
58	${}^{6}\chi^{v}$	Connectivity for six bond paths
59	${}^{7}\chi^{v}_{c}$	Connectivity for clusters with six bond paths
60	${}^{8}\chi^{v}_{s}$	Connectivity for stars with six bond paths
61	${}^{7}\chi^{v}_{r}$	Connectivity for seven bond rings
62	${}^{7}\chi^{v}$	Connectivity for seven bond paths
63	${}^{8}\chi^{v}_{c}$	Connectivity for clusters with seven bond paths
64	${}^{9}\chi^{v}_{s}$	Connectivity for stars with seven bond paths
65	${}^{8}\chi^{v}_{r}$	Connectivity for eight bond rings

[a] Symbols are taken from Stanton and Jurs [37].

$$\chi = \sum_{i=1}^{p} \frac{1}{(m * n)^{1/2}}, \tag{5}$$

where *p* is the number of bonds in the carbon skeleton, and *m* and *n* are the valencies of the two atoms in a given bond. The following equation gives an example calculation for *n*-butane, which contains two bonds between a primary and secondary carbon and one bond between two secondary carbons:

$$\chi = \frac{1}{(1*2)^{1/2}} + \frac{1}{(2*2)^{1/2}} + \frac{1}{(1*2)^{1/2}} = 1.914. \tag{6}$$

This descriptor is based on principles from graph theory and was used to build successful models for various physical properties such as Kováts indices, boiling points, and enthalpies of formation. Kier and Hall greatly expanded Randić's branching index to allow heteroatoms, carbons with hybridizations other than just sp^3, and extended paths other than just a two-atom bond [25, 26]. The resulting class of molecular connectivity descriptors is termed the valence-corrected connectivity descriptor and is denoted by χ^v. The calculated valency for a heteroatom is the number of attached nonhydrogen atoms plus the number of pi and lone pair electrons. An alcohol oxygen, for example, has a value of 5—a sum of 1 for the sigma bond and 4 for the lone-pair electrons. For a multiply bonded carbon, the valency is calculated as the number of attached bonds excluding hydrogens. For example, a carbon in a carbon-carbon double bond has a value of 2. The extension of the molecular connectivity descriptor to paths of greater than length 2 is illustrated by the following equation in which the summation occurs over all three-atom paths.

$${}^{2}\chi^{v} = \sum_{i=1}^{t} \frac{1}{(\mathrm{m} * \mathrm{n} * \mathrm{p})^{1/2}} \tag{7}$$

where *m*, *n*, and *p* are the valencies of the three atoms in the path, *t* is the number of three-atom paths in the molecule excluding hydrogens, and the superscript 2 denotes that fact that there are two bonds in the paths over which the connectivity is calculated. The molecular connectivity descriptor has not only been expanded to include paths of all lengths, but also includes other connectivity arrangements such as clusters (c), stars (s), and rings (r). A cluster is a path in which one atom has one branch. A star is a path in which one atom has two branches. A ring is a path in which two nonadjacent atoms in the path are connected to each other. When considering an arbitrary substructural fragment, the atoms not involved in that fragment are ignored. For example, a path of length 3 can have a branching atom, but, if one is looking for all paths of length 3, that particular path is still counted as a valid path of length 3. The most involved aspect of the molecular connectivity calculation is the enumeration of all the paths, clusters, stars, and rings of a specific atom count. A discussion of some software that has been written to do this is given in Appendix A.

The only purely electronic descriptors used in this work are the dipole moment, the polarizability, and four partial charge descriptors (descriptors 34, 35, 38, 39). The dipole moment and polarizability were calculated using *Gaussian 94* as described in the experimental section. Partial atomic charges were also calculated using *Gaussian 94* and were combined with surface area for the CPSA descriptors. The CPSA descriptors allow the distribution of charge in a molecule to be described on an atom-per-atom basis. These descriptors require the calculation of the surface area of a molecule and, more specifically, the surface area associated with each atom in the molecule. This is a nontrivial calculation. Stanton and Jurs reference Pearlman [35] for his method for the calculation of surface area. In Pearlman's article, several methods are discussed for the calculation of surface area. One of these methods has been adapted and implemented for the work described in this paper and is discussed in Appendix B.

3.2 Neural Networks. There are many articles in the chemical literature that review neural networks and even more that report the application of neural networks to some particular problem. Neural networks can be applied to a very wide range of problems, such as classification and model building, and have been proven to be universal function approximaters. The history of neural networks is reported to reach back to the 1940s and 1950s and has been discussed to some extent in a number of sources [41–43]. Neural networks achieved prominence in the 1960s until a noted researcher in the artificial intelligence community, Marvin Minksy, proved that the neural networks of that era, perceptrons, were incapable of classifying systems that are not linearly separable [44]. Perceptrons are neural networks that have an input layer and an output layer and use linear activation functions, and they are equivalent to linear learning machines. Neural networks entered a relatively dormant stage until the 1980s when research was published that overcame the limitations of the perceptrons of the 1960s. In 1982, Hopfield published a paper in which nonlinear transfer functions were introduced [45]. In 1986, Rumelhart, Hinton, and Williams published a key paper [46] in which the very popular "back propagation" training algorithm was introduced. Rumelhart's method incorporates the generalized delta rule as part of the training process and is discussed in some detail later in this paper. Several reviews of neural networks have been published in the chemical literature [41–42, 47–51]. In addition, a number of application papers do a good job of explaining the basic concepts of neural networks [52, 53]. A book by Zupan and Gasteiger [54] has a very good explanation of the generalized delta rule and the accompanying back propagation training algorithm as introduced by Rumelhart. A neural network frequently asked questions (FAQ) site maintained by Warren Sarle is a very good source of information and contains recommended reading resources [55]. Two very useful books by Timothy Masters provide good explanations and source code in C++ for the key aspects of multilayer feed-forward neural networks [43, 56].

There are a great many types of neural networks based on the architecture and training algorithm used. Lists of many of these types along with varying degrees of description can be found in a number of references [41, 50, 53]. The most frequently used general-purpose neural network and the network used in this work is the multilayer feed-forward neural network. This type of neural network generally uses supervised training in which both the input and output values of the calibration set members used to train the neural network are known. The process of training the neural network consists of many training cycles in which each member of the calibration set is presented to the neural network. The input values for a given structure are propagated forward through the neural network to the output node at which point the calculated output is compared to the actual output (the dielectric constant in this case) and an error is calculated. This error is propagated backward through the neural network, and the amount of error associated with each weight value is calculated. The training algorithm then uses this information to adjust each of the weight values in a manner that reduces the error of the calculated dielectric constant for the current member of the calibration set. By incrementally adjusting the weight values of the neural network to accurately calculate the dielectric constants of the calibration set structures, a generalized model is developed that will predict the dielectric constant of any compound that shares similar structural characteristics with the compounds in the calibration set. The remainder of this section discusses the details of the operation of multilayer feed-forward neural networks. Some of the key parameters associated with these neural networks are listed in Table 2.

Table 2. Key Parameters in Neural Networks

Number of hidden layers (generally one)
Number of nodes in hidden layer
Use of bias nodes
Scaling of inputs
Method used to calculate net input to a given node
Activation function
Error function
Method for optimization of weights
Determination of stopping point for traning

3.2.1 Forward Propagation. The first step in the neural network process is the forward propagation of information through the network. It is this forward propagation that is responsible for the feed-forward label in the type of neural network used in this paper. Figure 1 is given to illustrate the network and the forward-propagation process. The circles in layer 0 represent the input variables to the model (the molecular descriptors). The circle in layer 2 represents the output variable (the dielectric constant). For some applications, such as classification, there are multiple nodes in the output layer. The circles in layer 1 represent nodes in a hidden layer. The hidden layer provides extra terms for use in building the model. Although a hidden layer is not required, it is normally necessary to build a good model. Generally, only one hidden layer is used, but more can be used if the relationship being modeled is highly complex. One node in each layer except for the

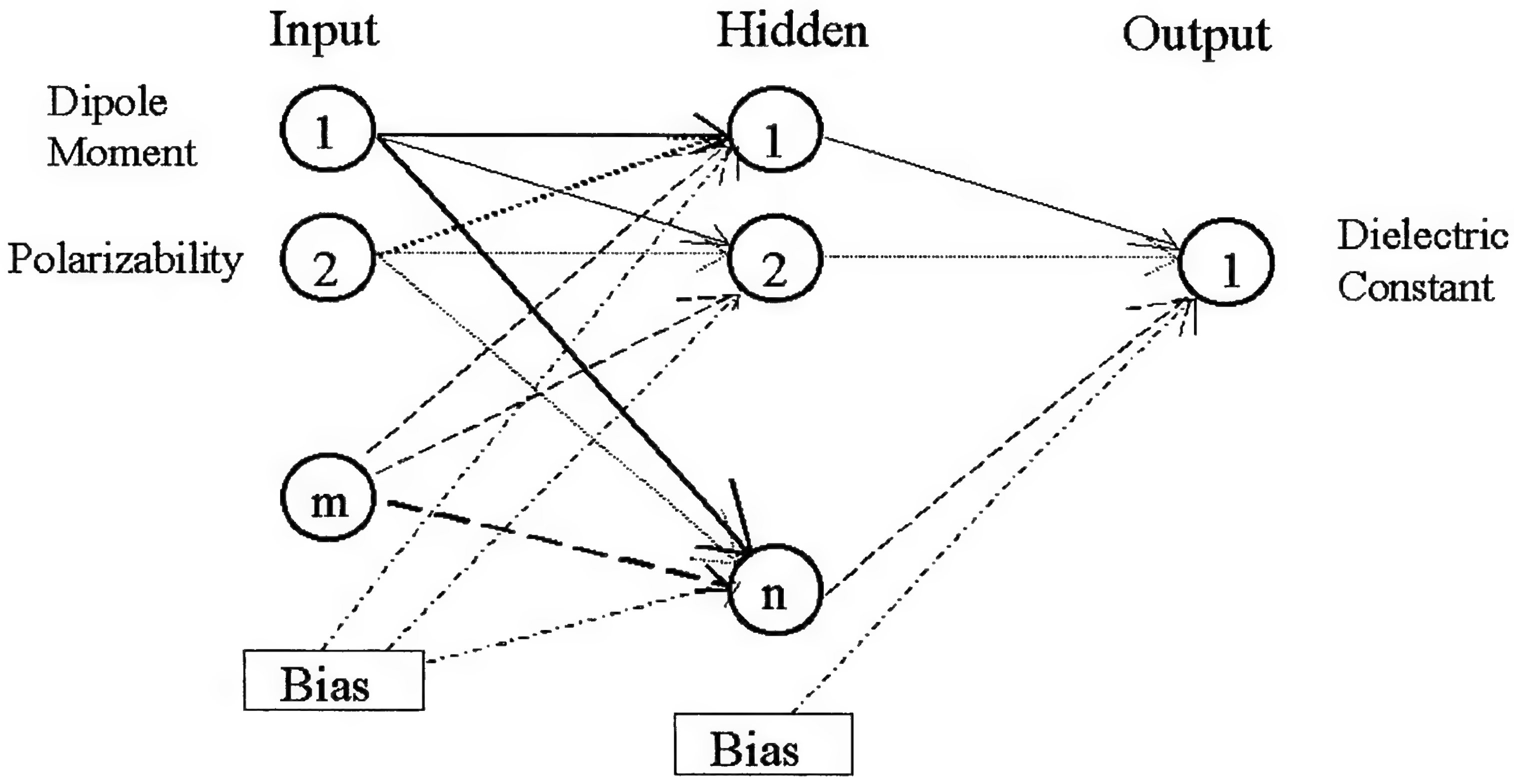


Figure 1. Representation of a Three-Layer Neural Network.

output layer has been labeled a bias node. Bias is analogous to a constant offset in least-squares regression and gives added flexibility to the neural network. One can think of the bias node as an extra source of input for each layer in the neural network. The bias node is generally assigned an output value of 1.0 and differs from other nodes in that nodes in a given layer do not propagate information forward to the bias node in the next layer. Every node in a given layer is connected to every nonbias node in the next layer, and each of these connections is represented by an arrow in Figure 1. A neural network in which there is a connection between every pair of nodes in adjacent layers is termed a fully connected neural network. Associated with each connection is a weight value, W_{ij}^{k}, where k denotes the second layer in the connection, i is the label of the node in the first layer of the connection, and j is the label of the node in the second layer of the connection. The weight values are initially randomly assigned as small real numbers. Forward propagation occurs by taking the output from every node in a given layer and multiplying that output by its associated weight value for each connection to a given node in the next layer. These products are summed together, and this net value is the combined input to a specific node in the next layer. Consider, for example, the contribution from the nodes in layer 0 to node 1 in layer 1. The net input to this node for a given member of the training set is

$$Net_1{}^1 = \sum_{i=1}^{m} W_{i1}{}^1 * X_i + \theta, \qquad (8)$$

where the superscript of 1 for *Net* refers to the layer to which the input is going, the subscript of 1 for *Net* refers to the node to which the input is going, X_i is the value of the ith input (molecular descriptor in this case) for that member of the training set, θ is the weight value associated with the bias node, and $W_{i1}{}^1$ is the weight value for the connection of the ith node in layer 0 to the first node in layer 1. The biological analog is the fact that a neuron receives an input from all of the neurons connected to it. The summed value, $Net_1{}^1$, is acted on by a function called the activation function or the transfer function. As an example, the value output from node 1 in the hidden layer is

$$Out_1{}^1 = f\left(Net_1{}^1\right). \qquad (9)$$

The purpose of this function is to allow the calibration model to model any nonlinearities present in the relationship between the molecular descriptors and the dielectric constant. The biological analog to the activation function is the fact that a certain amount of electrical stimulus must reach a neuron from connecting neurons for that neuron to fire a signal to the next layer of neurons. This biological activation function is a step function. Two very popular nonlinear transfer functions are the sigmoidal transfer function and the hyperbolic tangent function. Most literature sources report that the choice of activation function does not greatly affect the results of the neural network as long as a nonlinear function is used. The inputs to a neural network are typically scaled to fit the range appropriate for the activation function ($Y = 0.0$ to 1.0 for sigmoidal; $Y = -1.0$ to 1.0 for hyperbolic tangent). The scaling is carried out independently on the set of values (one value for each member of the calibration set) for each independent variable (molecular descriptor) and dependent variable (dielectric constant) in the model. Although scaling is not considered necessary for the successful training of a neural network, it does greatly speed up the training process and is generally employed. A typical method of scaling and the method used in this work is to simply subtract the midpoint of the set of numbers from the number to be scaled and then to divide the resultant by the range of the set of numbers [52]. Another option would be to use the mean and standard deviation of the set of numbers for scaling [57].

This process of summing inputs and applying a transfer function is repeated for each of the nodes in the hidden layer. The resulting output values of the hidden layer nodes are used as input values for the node in layer 2. The process is repeated and an output value is calculated for this node. This calculated output is compared to the actual dielectric constant after the appropriate scaling, and the difference is the error of the current member of the calibration set for the current set of weight values for the connections in the neural network. The error can be calculated in various ways, and the error function chosen will have an effect on the results obtained by the neural network. One can use an absolute value of the difference, or a squared value of the difference, or a relative value in which the difference is divided by the true value. If there are multiple output nodes, a mean calculation of the error values must be used in which the error associated with each node in the output layer is included. In this work, the squared value of the difference at the single output node is used.

3.2.2 Backward Propagation of Error. The second step in the neural network process is the backward propagation of the error. The result of this step is that the error is distributed proportionally across the neural network with the nodes contributing the largest input (having the most significance) accepting the largest amount of the error. Mathematically, the amount of error associated with a given neural network connection is the partial derivative of the error with respect to the weight associated with that connection. The mechanism by which this procedure is accomplished is by use of the chain rule from calculus, which states

$$\frac{d}{dx}[f(u)] = \frac{d}{du}[f(u)] * \frac{du}{dx}. \tag{10}$$

For this application, $f(u)$ is the error such that

$$f(u) = E^k\left(output_j^k\left(net_j^k\left(w_{ij}^k\right)\right)\right), \tag{11}$$

where the error, E, is the error associated with the output of the jth node of the kth layer and the input to that jth node is from the ith node in layer $k - 1$. Thus, the error associated with a given connection from layer $k - 1$ to layer k is a function of the output in layer k, which is a function of the transfer function, which is a function of the input from the ith node in layer $k - 1$. Using multiple applications of the chain rule, the partial derivative of this function is:

$$\frac{\partial E^k}{\partial W_{ij}^k} = \left(\frac{\partial E^k}{\partial Out_j^k}\right) * \left(\frac{\partial Out_j^k}{\partial Net_j^k}\right) * \left(\frac{\partial Net_j^k}{\partial W_{ij}^k}\right). \tag{12}$$

A thorough explanation of the details for the derivations of each of these terms can be found in chapter 8 of the book by Zupan and Gasteiger [54]. The results of these derivations are given here. The partial derivative of Net_j^k with respect to W_{ij}^k is Out_j^{k-1}. Since Out_j^k is a result of the application of the transfer function, the partial derivative of Out_j^k with respect to Net_j^k is the derivative of the transfer function that was used, evaluated for the value Net_j^k. As an example, the derivative for the sigmoidal transfer function is $Out_j^k(1 - Out_j^k)$. The partial derivative of E^k with respect to Out_j^k depends upon k (the layer). If the layer is the output layer, the partial derivative is $-2(y_j - Out_j^k)$, where y_j is the actual value associated with the output node j in the output layer and Out_j^k is the calculated value for the output node j in the output layer. If the layer is a hidden layer, the calculation of the partial derivative is less straightforward because the error is not calculated directly. If the assumption is made that the error is distributed evenly across all nodes in the hidden layer, the partial derivative is

$$\frac{\partial E^k}{\partial Out_i^k} = \sum_{j=1}^{m} \delta_j^{k+1} W_{ij}^{k+1}, \tag{13}$$

where m is the number of nodes in layer $k + 1$, i is the node of interest in the hidden layer, and δ_j^{k+1} is the error that has been propagated backward from layer $k + 1$ and is given by

$$\delta_j^{k+1} = \left(\frac{\partial E^{k+1}}{\partial Out_j^{k+1}}\right)\left(\frac{\partial Out_j^{k+1}}{\partial Net_j^{k+1}}\right), \tag{14}$$

where both terms in this equation are the calculated values from the output layer.

Consider the back propagation to proceed in multiple steps for a neural network with an output layer and one hidden layer. The first step calculates the error associated with each connection to the node in the output layer. The second step propagates the error from the output layer back to the hidden layer in the form of delta values. These delta values allow the error associated with each connection to the hidden layer to be calculated. The full equation for the output layer for the case in which a sigmoidal transfer function is used is

$$\frac{\partial E^k}{\partial W_{ij}^k} - 2\left(y_j - Out_j^k\right)Out_j^k\left(1 - Out_j^k\right)Out_i^{k-1}, \tag{15}$$

where *k* is the output layer, *i* is the node in the hidden layer for the connection, and *j* is the node in the output layer for the connection. The full equation for the hidden layer is

$$\frac{\partial E^{k}}{\partial W_{ij}^{k}} = \left(\sum_{p=1}^{m} \delta_{p}^{k+1} W_{jp}^{k+1} \right) Out_{j}^{k} \left(1 - Out_{j}^{k}\right) Out_{i}^{k-1}, \tag{16}$$

where *k* is the hidden layer, *i* is the node in the input layer for the connection, *j* is the node in the hidden layer for the connection, and *m* is the number of nodes in the output layer. Using these equations, a partial derivative of the error can be calculated for each of the connections in the neural network.

3.2.3 Optimization of the Weights. The third step in the neural network process is the adjustment of the weight values based upon the distributed error values, $\partial E/\partial W_{ij}^{k}$, as calculated in the previous section. The weights can be adjusted after the presentation of a single member of the calibration set to the neural network or after all members of the training set have been presented to the neural network. In the work presented in this paper, the latter method is used. The distributed error value for a given connection is the sum of the error values for each of the members of the calibration set for that connection. The method by which these distributed errors are used to adjust the weights is called the training method. A helpful way to visualize the weight adjustment process is by the aid of a response surface as illustrated in Figure 2. The *x* and *y* axes represent two connections in the neural network. The weight values for each connection may be varied across a range of values. An actual neural network would be represented by an *n*-dimensional response surface rather than a two-dimensional (2-D) response surface where *n* is the number of connections in the neural network. The *z* axis represents the error associated with a given state of the neural network, where the error can be calculated using any of the error functions described previously. The collection of weight values for a given state can be thought of as a weight vector. Associated with each point on the response surface is the vector of the partial derivatives of the error with respect to each weight as obtained by the back-propagation process.

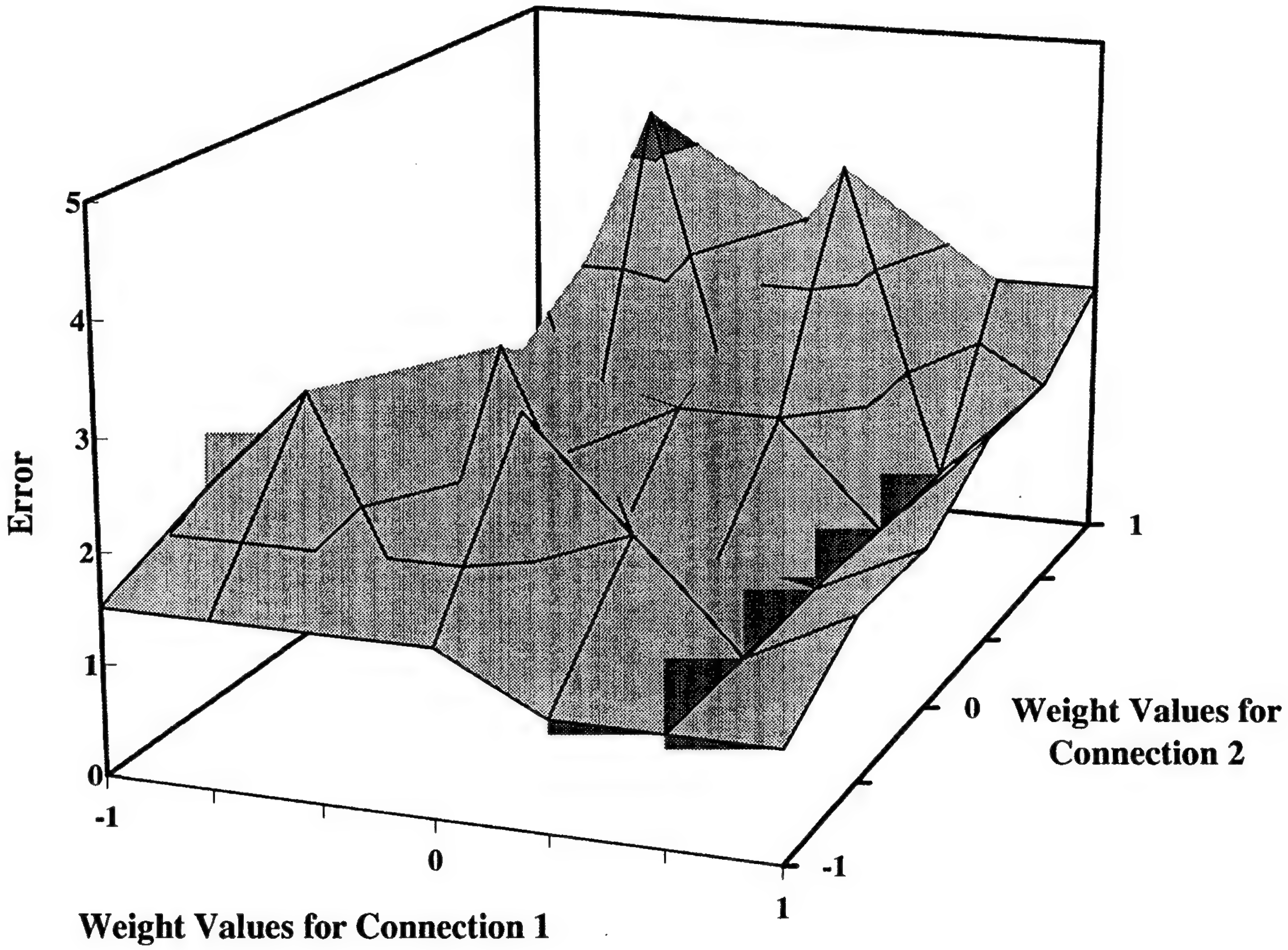


Figure 2. Illustration of a Hypothetical Response Surface for a Two-Connection Neural Network.

The response surface imagery previously described frames the search for the best set of weight values as an optimization problem. The process of optimization is a major subject area about which much has been written [58, 59]. There are two choices attendant with any optimization process. The first choice is to select an algorithm that will find the closest local minimum, and the second choice is to select an algorithm that will find the global minimum. Some of the techniques used to find global optima include simplex optimization, simulated annealing, and genetic algorithms [60–67]. While using a global optimum technique would be the ideal solution, there are many parameters that must be set and tweaked to use a global technique. In addition, there is no guarantee that a global minimum will be found. Masters [43, 56] covers the use of simulated annealing for weight optimization in neural networks, and Jurs and coworkers [68, 69] have reported the use of this technique in several application papers.

The search for a local minima is a much simpler problem, and the majority of training algorithms associated with neural networks are local optimization routines. If one repeats the local optimization numerous times at different randomly selected points on the response surface and chooses the best local minimum from the repeated runs, a good local minimum approximating the global minimum should be found. The process by which the best run is selected is described in a later section. The fact that neural networks have multiple global minima due to the symmetries present increases the probability that a good local minimum will be found. The most popular local minimization algorithm is the steepest descent method developed by Rumelhart [46] and incorrectly referred to as back propagation. This is a misnomer as back propagation should refer to the propagation of the error backward through the neural network and not to the algorithm used to adjust the weights.

The first step in any local optimization procedure is to select the direction on the response surface in which to move away from the current position. The direction chosen by the steepest descent method is the negative of the gradient, which is calculated in the back-propagation step. This gradient is simply the vector of partial derivatives of the error associated with each point on the response surface. The second step in a local optimization procedure is the choice of how big a step to take from the current position on the response surface. The steepest descent method uses a fixed value called the learning rate, which is set by the user. Some versions of this method allow the user to change the learning rate at fixed points throughout the training process. Most implementations of Rumelhart's steepest descent method add a second parameter, momentum. The momentum is a constant that is multiplied by the previous adjustment to the weight vector. The resulting value is added to the new weight value and helps to keep the optimization moving in a constant direction unless a dramatically new direction is introduced by the gradient. The steepest descent method for training neural networks as introduced by Rumelhart is a useful method, but it is very inefficient when compared to other more advanced methods that have been introduced since that time. It is probably still so widely used because it is such a familiar and historically important algorithm.

Local optimization can be divided into three major classes: nonderivative methods, first derivative (gradient) methods, and second derivative methods. Nonderivative methods use only information from the function being optimized and are relatively inaccurate and unreliable and, hence, are not generally used. First derivative methods use gradient information calculated from the function to determine the search direction on the response surface for the optimization. The steepest descent method falls into this class. Conjugate gradients are another valuable method belonging to this class. The second derivative methods use the second derivatives of the function (the Hessian) to determine the search direction. Discrete Newton, truncated Newton, and quasi-Newton methods fall into this class. Another method, the Levenberg-Marquadt method, can also be classified as a second derivative method. These three classes of local optimization methods are discussed by Schlick in a review article [70] and in the optimization texts already mentioned. In addition, Masters [43, 56] goes into great detail explaining the conjugate gradient method and the Levenberg-Marquardt method and includes computer code for these methods as well.

The algorithm used in this work is a quasi-Newton method developed by Broyden [71], Fletcher [72], Goldfarb [73], and Shanno [74]. It has been used successfully in a number of papers by Jurs and coworkers [75–77]. It is a much more efficient method than Rumelhart's steepest descent method. One reason for the utility of this method is that the search direction is determined using second derivative information rather than just gradient information. A second reason is that a line search is used to determine the step size rather than using a fixed step size. Historically, second derivative methods have not been feasible because of the difficulty of calculating the Hessian and its inverse and because of the large amount of memory required. The Broyden, Fletcher, Goldfarb, and Shanno (BFGS) method overcomes both of these problems. The first problem is overcome by the use of an approximation to the inverse of the Hessian rather than explicitly calculating this matrix. This approximation is calculated using only gradients and function information. The second problem is overcome by the use of an update procedure in which the approximated Hessian is updated after each step in the optimization process. Only a small number of the most recent updates (three to seven) are needed to calculate the most current inverse of the Hessian. Therefore, only the most recent updates need to be stored in memory and older updates are deleted. An added benefit is that as the updating process progresses, the approximation to the Hessian becomes increasingly more accurate. Nocedal [78] wrote an important paper describing the implementation of the BFGS method [78]. In addition, there is information on a website including tutorials on optimization and source code [79]. A key equation in this approach is the Newton equation:

$$\mathbf{B}_k \mathbf{d}_k = -\nabla f(\mathbf{X}_k), \tag{17}$$

where k is the current iteration, $\mathbf{B}_k$ is the Hessian, $\mathbf{d}_k$ is the search direction, and $\nabla f(\mathbf{X}_k)$ is the gradient at $\mathbf{X}_k$. This equation can be solved for $\mathbf{d}_k$ as follows:

$$\mathbf{d}_k = -\mathbf{H}_k \nabla f(\mathbf{X}_k), \tag{18}$$

where $\mathbf{H}_k$ is the inverse of the Hessian. The formula that was developed to update the inverse Hessian without explicitly calculating the Hessian or its inverse is

$$\mathbf{H}_{k+1} = \left(\mathbf{I} - \frac{\mathbf{S}_k \mathbf{Y}_k^T}{\mathbf{Y}_k^T \mathbf{S}_k}\right) \mathbf{H}_k \left(\mathbf{I} - \frac{\mathbf{Y}_k \mathbf{S}_k^T}{\mathbf{Y}_k^T \mathbf{S}_k}\right) + \frac{\mathbf{S}_k \mathbf{S}_k^T}{\mathbf{Y}_k^T \mathbf{S}_k}, \tag{19}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{S}_k$ is $\mathbf{X}_{k+1} - \mathbf{X}_k$, $\mathbf{Y}_k$ is $\nabla f(\mathbf{X}_{k+1}) - \nabla f(\mathbf{X}_k)$, $\mathbf{H}_{k+1}$ is the current approximation to the inverse of the Hessian, and $\mathbf{H}_k$ is the approximation to the inverse of the

Hessian from the previous iteration. This formula lends itself to a recursive implementation. The first step is the approximation of the initial inverse of the Hessian, $\mathbf{H}_0$. An identity matrix is generally used for $\mathbf{H}_0$. Given $\mathbf{H}_0$, $\mathbf{H}_1$ can be calculated. Given $\mathbf{H}_1$, $\mathbf{H}_2$ can be calculated and so on. The article by Nocedal gives a clear step-by-step presentation of this updating procedure. After each update of $\mathbf{H}_k$, a line search is performed in the search direction obtained by $\mathbf{H}_k$ to find the correct step size, α. In its simplest form, a line search uses a series of carefully chosen guesses to bound the minimum in one dimension and refine that range to some degree of precision, at which point the midpoint of the range is selected as the desired step size. A clear explanation of the line search algorithm along with source code is given by Masters [43]. This code was translated to FORTRAN and used in the neural network software developed for this research. Given the direction, $\mathbf{d}_k$, and the step size, α, the new value for $\mathbf{X}_{k+1}$ is

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \alpha \mathbf{d}_k . \qquad (20)$$

Each iteration of the BFGS algorithm represents one training cycle. It was found that the BFGS method requires a small fraction of the number of training cycles needed by back propagation. There are more function evaluations per cycle for the BFGS method then back propagation, but the amount of training time is still significantly smaller for BFGS method then for back propagation.

An important issue in training a neural network is the stopping point for the training. One method is to keep training until the calibration error drops below a certain point. The danger with this method is that a model developed by a neural network can be overtrained, at which point the model loses its ability to make accurate predictions for compounds that are not in the calibration set. The currently accepted procedure that avoids overtraining is the use of a monitoring set. The error for a set of compounds that are not members of the calibration set is calculated at regular intervals throughout the training. As the quality of the model improves, the error associated with the monitoring set decreases. At some point in the training, the model begins to overtrain. At this point, the error associated with the monitoring set begins to increase. When this increase in monitoring error is detected, training is halted and the neural network associated with the minimum monitoring set error is chosen as the appropriately trained neural network. It was found that the monitoring set error did not always reach a minimum and then gradually and smoothly become larger with more training cycles. However, an acceptable solution was found using the monitoring set. Every neural network was trained for 250 training cycles using the BFGS method. Generally, the training set error rapidly decreases and reaches a relatively constant level by 50 iterations. A search then began for the iteration with the lowest training set error over the 250 iterations with the condition that the corresponding monitoring set error not exceed 110% of the average monitoring set error between 50 and 80 iterations. Given an appropriate ending point for training, a third independent set of compounds, the test set (validation set), is then used to test the resulting neural network.

3.3 Model Building. Two important and interconnected steps in the model building process are the selection of the independent variables and the selection of the calibration set. One can think of the set of independent variables as the set of coordinate axes that defines the data space into which the members of the training set (calibration set), monitoring set, and test set fall. These three subsets form the universe of data points. It is important that the data points in the universe evenly cover the entire data space over which the calibration model is built and over which one wishes to make predictions. An additional and opposing goal is to make this data space as large as possible so that predictions can be made over as wide a range as possible. Generally, one must limit the size of the data space to ensure the development of accurate calibration models.

Normally, one does not know which independent variables are truly necessary for the prediction of the dependent variable. A set of compounds can be selected, which evenly cover the data space, but, if the coordinate axes of that data space do not collectively correlate well with the dependent variable, the model developed will be poor. In this work, a number of models have been created using different sets of independent variables and the model has been selected, which gives the lowest error for the calibration set and the test set. To allow a valid comparison of the models, one should use the same calibration set and the same test set. There is a conflict, however, because the distribution of compounds in the data space may be even for some of the models, but may be uneven for other models. A compromise must be reached, where the universe of compounds is limited so that the compounds in the calibration set are distributed as evenly as possible in each of the data spaces as defined by the coordinate axes for each model.

The set of 540 compounds, as described in the experimental section, have dielectric constant values ranging from 1 to 185. Figure 3 shows the distribution of the compounds based on dielectric constant. A total of 67.6% of the compounds falls within the range from 1 to 10. An additional 29.4% of the compounds fall within the range from 10 to 40. Only 16 compounds have a value greater than 40. Some preliminary experiments gave very poor results for models built over the full dielectric constant range. The best results were obtained for models built over the range of 1 to 10. By restricting models to this range, however, it would not be possible to predict dielectric constants for many of the compounds in which there was an interest. The preliminary models built over the dielectric constant range from 1 to 40 gave test set results that were acceptable. Therefore, all of the models reported in this research were developed from a calibration set in which no compounds with dielectric constants greater than 40 were used. From a theoretical viewpoint, it is more appealing to limit the compounds in the calibration set based on the values of the independent variables, but, from a practical point of view, it made sense to limit the calibration set based on the dependent variable. Of the 524 compounds remaining, only 27 contain flourine, sulfur, or iodine. These compounds were removed, and a total of 497 compounds remain.

Given the final set of compounds selected for model building, the next step is the division of these compounds into a training set, a monitoring set, and a test set. The compounds for each training set were chosen so that the most diverse compounds in the universe were included. The monitoring set compounds were selected from the compounds not included in the training set, and the remaining compounds were used as the test set. The first step in this selection process for a given experiment is the performance of principal components analysis (PCA) to reduce the dimensionality of the data space from *m* independent variables to *n* principal components, where *n* is the number of principal components needed to explain 95% of the variance in the data. A maximum of 10 was allowed for *n* because of memory limitations. PCA is a data-reduction algorithm that is widely used and is described in a number of tutorials [80–81]. Each of the coordinate axes is partitioned into a number of equally spaced partitions, and the compounds

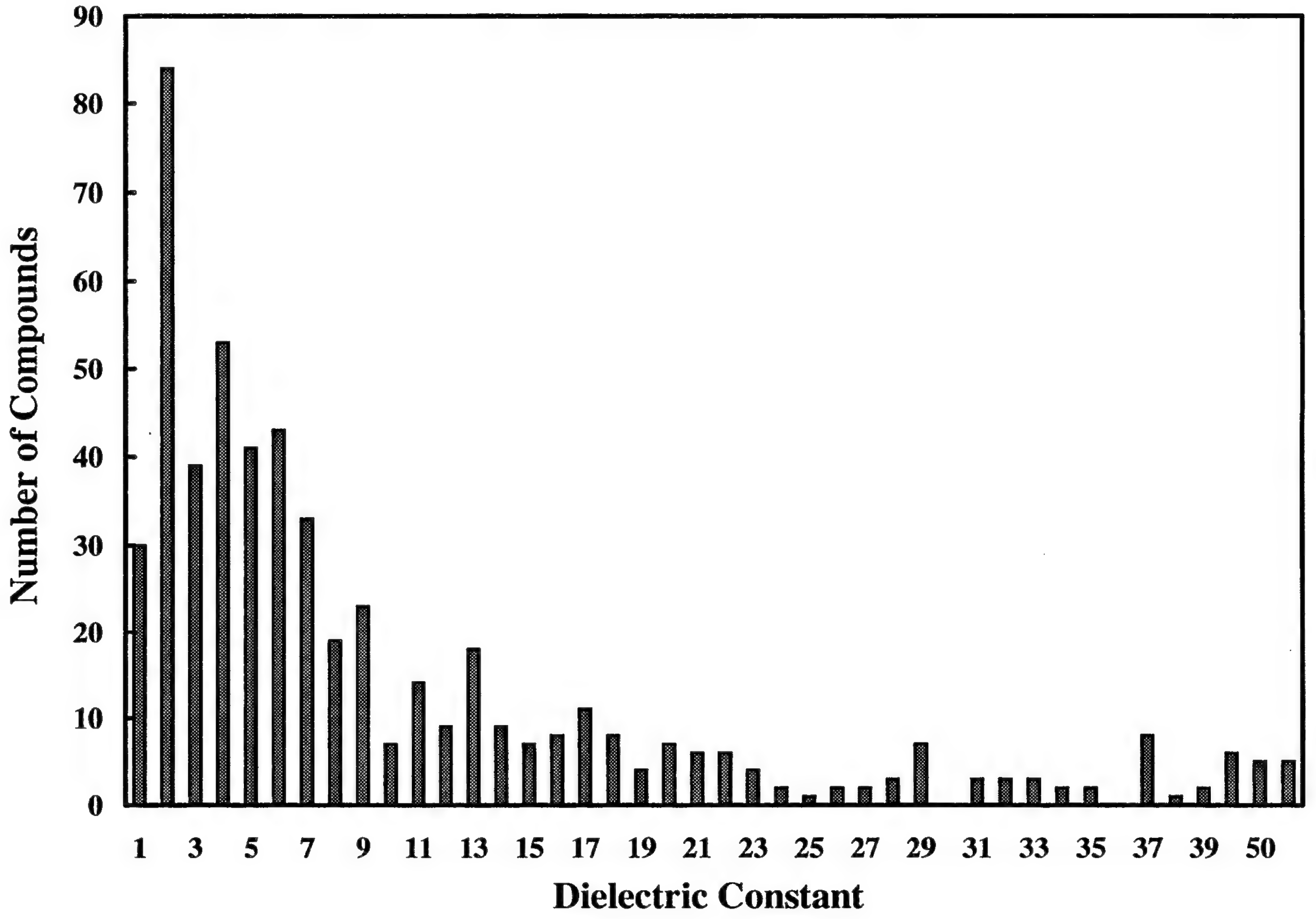


Figure 3. Distribution of the Dielectric Constants in One-Unit Intervals for a Set of 540 Compounds. The Last Three Categories Are >40 and ≤50, >50 and ≤60, >60.

in the universe are then mapped into the resulting n-dimensional grid, where n is the number of principal components selected. The size of the partition is selected so that the total number of occupied blocks is less than or equal to the number of compounds in the training set. The box in the n-dimensional grid into which each compound falls is recorded. The occupied boxes are then traversed, and the most centrally located point in each box is selected as a member of the current data set—training set or monitoring set. If after 10 traversals of the data space the desired number of compounds for the current data set is not found, that model is not built. The software used for this algorithm was a revised version of the software written by Carpenter and Small [82]. In all of the models reported in this work, 350 compounds are selected for the training set, 50 compounds are selected for the monitoring set, and the remaining 97 compounds are used for the test set. The drawback to this method is that the same set of compounds is not always chosen for the training set, monitoring set and test set for each of the models developed. This drawback was outweighed by the fact that this method results in the most representative sampling of the universe of data points for each set and by the fact that the combined members of the data sets remain constant.

3.4 Neural Network Parameters. The number of nodes in the hidden layer for each neural network is set to a total of six nodes, including one node for the bias. The number of nodes in the input layer is simply the number of molecular descriptors plus one node for the bias. The activation function for the hidden layer and the output layer is the hyperbolic tangent. The inputs are scaled to the range from -1.0 to 1.0. Each neural network is trained using the BFGS algorithm for a total of 250 training cycles with the endpoint being selected as discussed earlier. There are two ways that the training set error is reported. The method used in the training of the neural networks is to calculate the mean-squared error for the scaled values of the dielectric constants in the training set. The resulting reported values range from approximately 0.01 to 0.06. The monitoring errors are calculated in the same manner and range from approximately 0.01 to 0.10. The second method used to report the training set errors is to calculate the mean absolute value of the errors for the dielectric constants after they have been converted back to their normal range. These values range from approximately 2.5 to 6.0. The test set errors are also calculated and reported using this method and range in value from approximately 1.2 to 12.0. In several clearly marked cases, root-mean-square error values in this range are also reported for the test and training sets. In the following sections, the particular error calculation method used can be determined by the approximate magnitude of the error.

3.5 Exploratory Experiments. The majority of the work reported in this paper is a search for the set of independent variables that best correlate with dielectric constant. In an earlier section, a large set of potential independent variables (molecular descriptors) is described (Table 1). If the entire set of variables is used, the resulting model would be drastically overfit. Therefore, a small subset of descriptors must be selected from this pool of descriptors. There are a number of standard techniques that can be used to do this [21, 83, 84]. The first step used was the removal of descriptors that have a large percentage of compounds for which the value is 0. Descriptors 7, 10, 11, 44, 45, 48, 49, 52, 53, 56, 60, 61, 64, and 65 all have less than 40 (of 497) nonzero values and were removed from consideration. The second technique employed was the calculation of a pairwise correlation for every possible pair of descriptors in the pool of descriptors. If the correlation was greater than some percentage, one of the two descriptors was removed from consideration. For the 51 descriptors remaining after step 1, there are 1,275 pairs of descriptors (51 * 50/2). Of these pairs, seven have correlations greater than 97%. As a result, descriptors 20, 23, 28, 30, 32, 39, and 41 were also removed from consideration. From the remaining descriptors, a Gram-Schmidt orthogonalization procedure was implemented to choose the set of descriptors that best covers the range of information encoded in the descriptors. The descriptor that correlates best with the dielectric constant is chosen as the first descriptor in the subset. The second descriptor chosen is the descriptor with the maximum amount of orthogonal information to the first descriptor in the subset. The third descriptor contains the maximum amount of orthogonal information to the first two descriptors in the subset. This process continues until a set number of descriptors is chosen or until a given percentage of variance in the data is explained. This procedure was performed independently for the three sets of descriptors. All nine simple descriptors were chosen as final members of the first set of descriptors. Twelve of the CPSA descriptors were chosen from the second set of descriptors (13–16, 18, 19, 22, 25–27, 31, 33). Ten of the molecular connectivity descriptors were chosen from the third set of descriptors (40, 42, 46, 50, 54, 55, 57–59, 62). The orthogonalization was also carried out for the set of all descriptors and was used along with the information from the results of model building for the first three sets of descriptors to choose 16 descriptors from all three sets of descriptors for a fourth set of descriptors (1, 2, 3, 5, 6, 12, 16, 22, 25, 27, 33, 40, 42, 46, 59, 62).

For each of the 4 sets of descriptors, calibration models were built for every possible subset of descriptors starting with a minimum of 4 descriptors in a model and resulting in a total of 382 experiments for the set of simple descriptors, 848 experiments for the set of molecular connectivity descriptors, 3,797 experiments for the set of CPSA descriptors, and 64,839 experiments for the combined set of descriptors. Each neural network must be repeated numerous times with different randomly selected initial weight values, as discussed earlier, to allow the selection of a neural network with a good local optima. Due to the large number of experiments to be performed, each neural network is repeated only 10 times and the average error of these 10 runs is used to compare the results of these exploratory experiments.

The distributions of the models for each of the four sets of descriptors based on the training set and test set errors are given in Figures 4–7. Tables 3 and 4 summarize these distributions for the training sets and test sets, respectively. Column 2 lists the number of models in that descriptor category that were successfully built. Column 3 lists the error for the model with the lowest error, and column 4 lists the error for the model with the highest error, where the reported error is an average of the errors of the 10 neural network runs. It should be noted that the model with the lowest training set error is generally not the model with the lowest test set error. Column 5 is the mean error of all the models that were successfully built for that particular descriptor category, where each number contributing to the average is itself an average of the 10 neural network runs. The mean error in column 5 essentially marks the center of the distributions, as seen in Figures 4–7. The standard deviation essentially describes the spread of these distributions. One can clearly see from these figures and tables that the molecular connectivity descriptors give the worst models and the CPSA descriptors give the next-to-worst results. The results from the simple descriptors compared to the results from the combined descriptors are closer, but the combined descriptors do give models that are clearly better than the best models obtained with the simple descriptors. There are

Table 3. Training Set Results for Exploratory Experiments

Set	No. of Models	Minimum Error[a]	Maximum Error[b]	Mean Error[c]
Simple	359	3.15	5.89	4.05 ± 0.56
Chi	784	4.27	5.41	4.79 ± 0.97
CPSA	3,797	3.60	5.59	4.32 ± 0.50
Combined	64,756	2.57	5.47	3.64 ± 0.92

[a] Training set error for the model with the minimum training set error.

[b] Training set error for the model with the maximum training set error.

[c] Mean of the test set errors.

Table 4. Test Set Results for Exploratory Experiments

Set	No. of Models	Minimum Error[a]	Maximum Error[b]	Mean[c]
Simple	359	1.83	6.65	3.66 ± 0.91
Chi	784	6.46	14.82	8.25 ± 0.97
CPSA	3,797	2.83	6.62	4.13 ± 0.50
Combined	64,756	1.28	12.00	3.80 ± 0.92

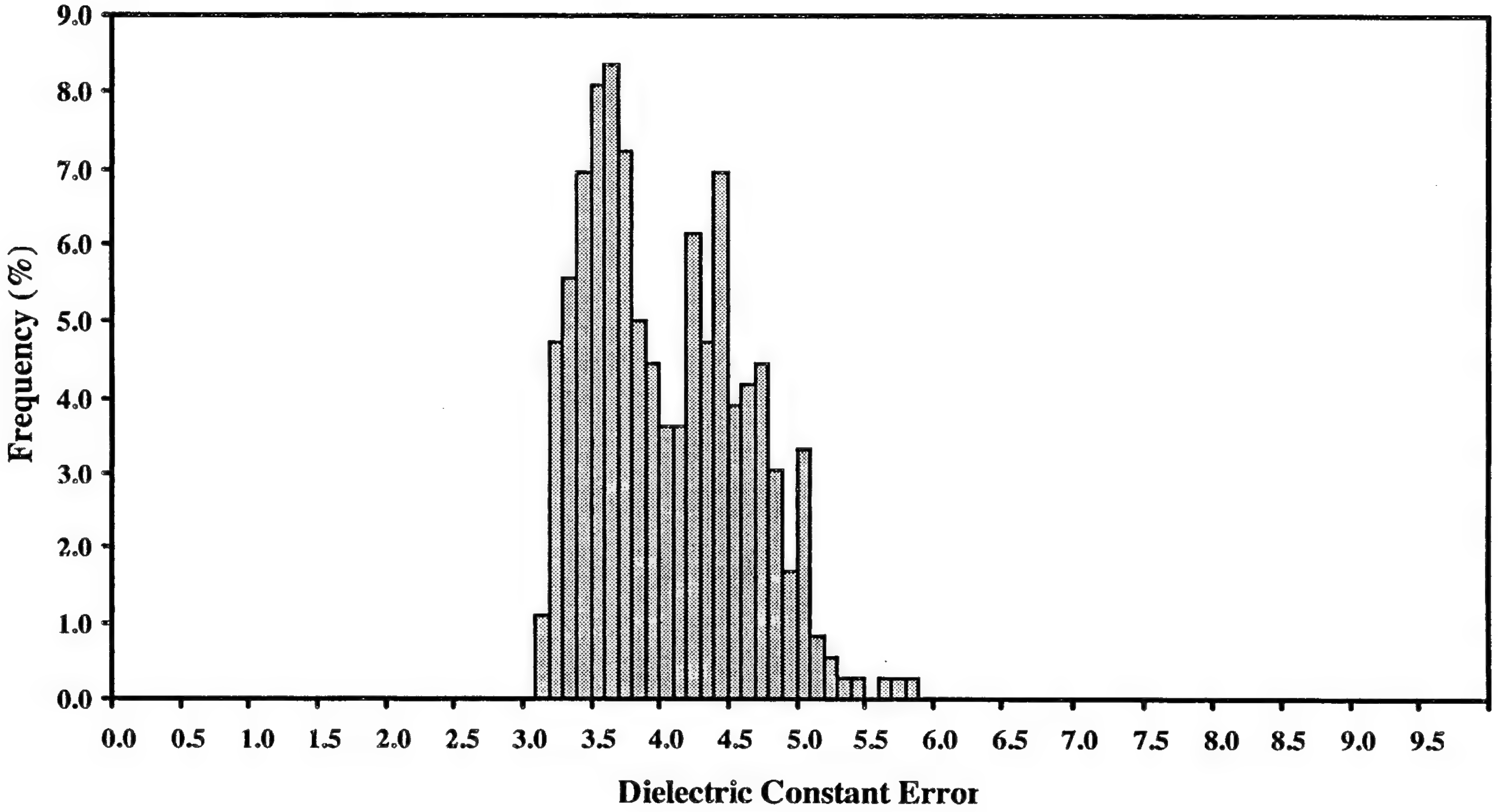
[a] Test set error for the model with the minimum test set error.

[b] Test set error for the model with the maximum test set error.

[c] Mean of the test set errors.

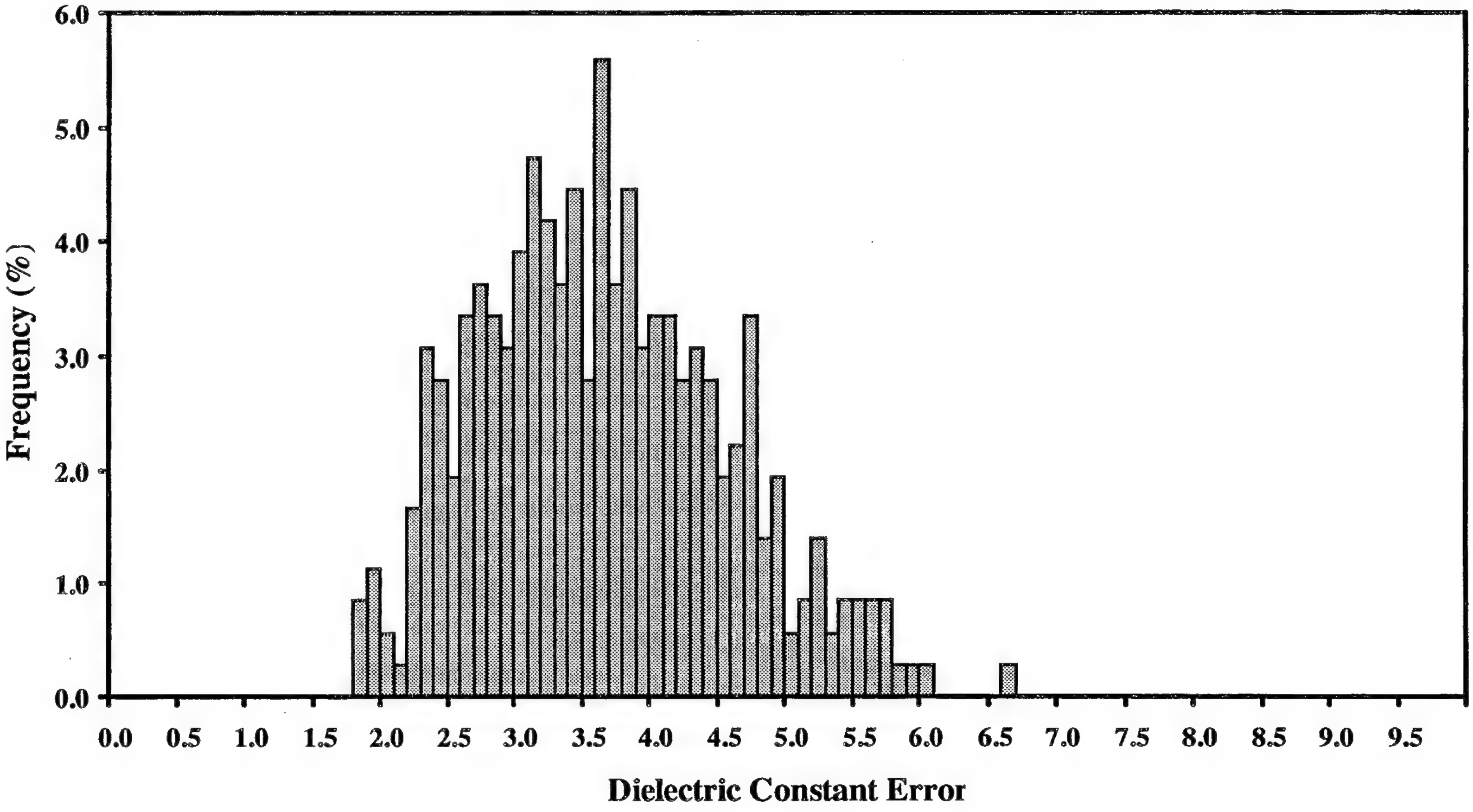
2,215 models using the combined descriptors, which have a training set error less than 3.4 and a test set error less than 2.4. For the simple descriptors, there are only 11 models that meet these criteria. Since the combined descriptor models give the best results, it was decided to look at only these models using the full neural network training procedure. The subset of 293 models was picked for which the training set error is less than 3.1 and the test set error is less than 2.1. Incidentally, there are no models using only the simple descriptors that meet these criteria.

3.6 Set of 293 Experiments. For each of the 293 models, 200 neural networks are trained with 200 different initial weight values. These 200 neural networks are ranked according to the training set error, and a list of the top 40 neural networks is created. This list of 40 neural networks is then ranked according to the monitoring set error, and the top neural network is selected as the neural



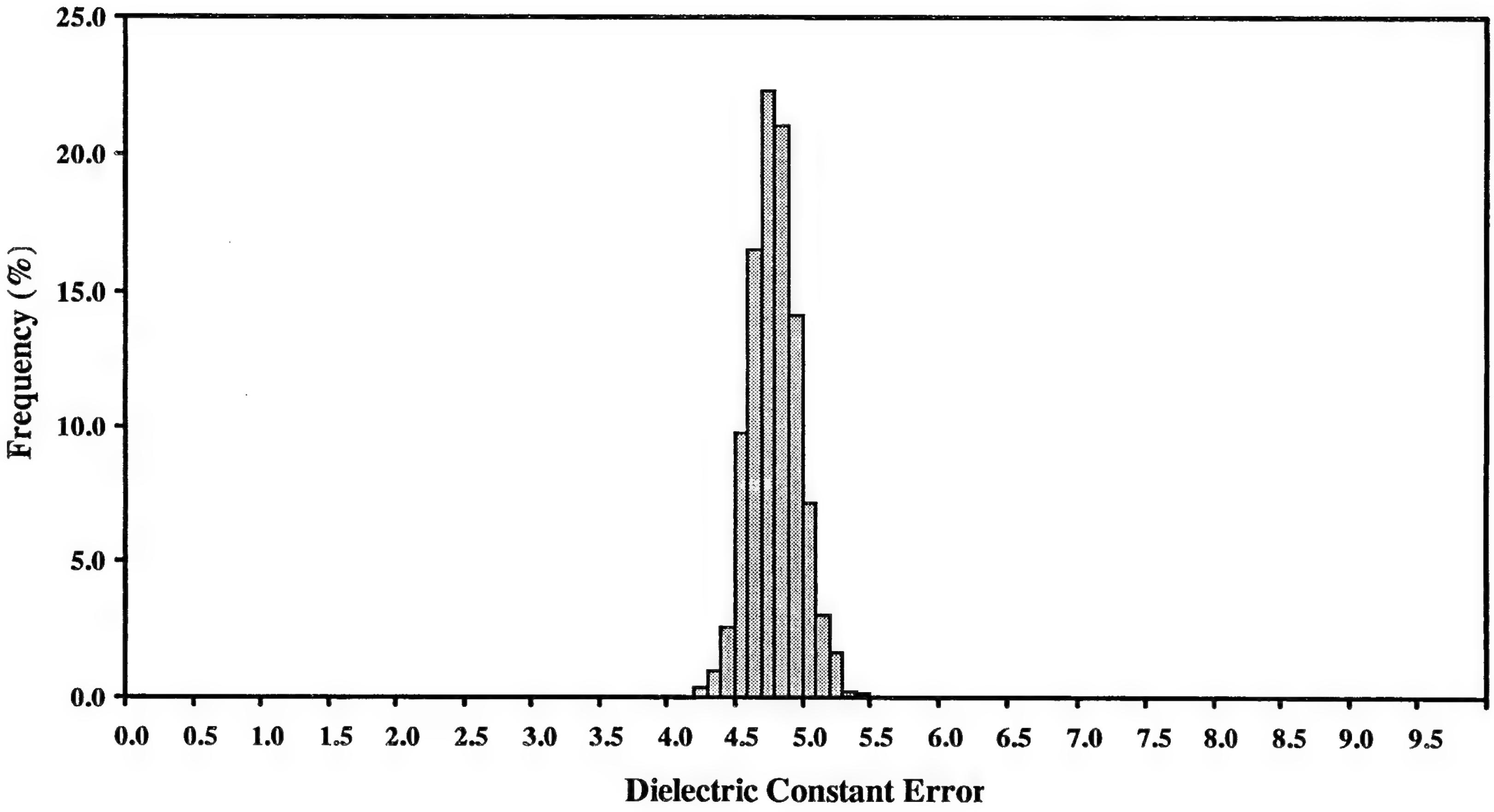
(a)

Figure 4. Distribution of the (a) Training Set and (b) Test Set Errors for the 359 Models Calculated Using the Simple Descriptors.



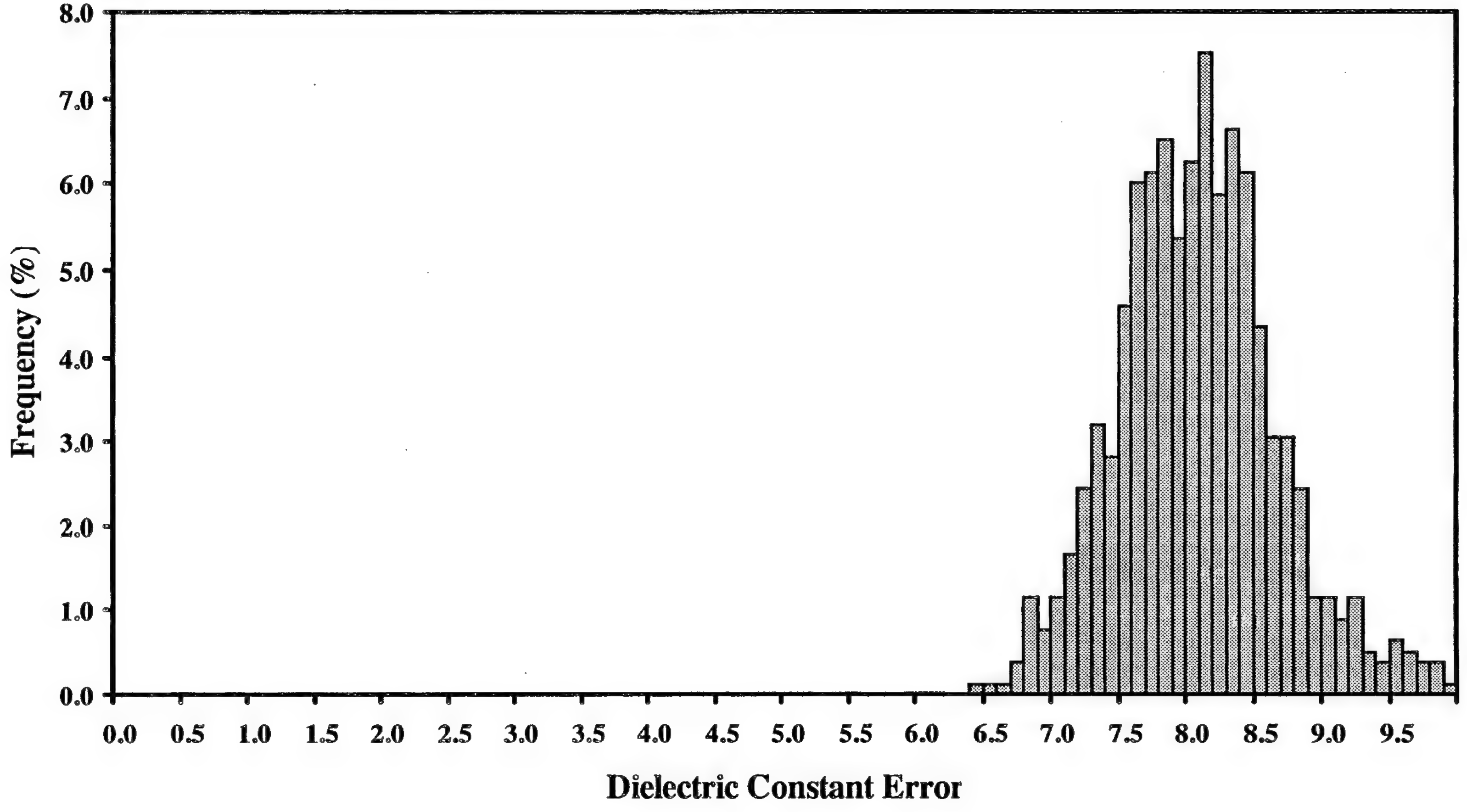
(b)

Figure 4. Distribution of the (a) Training Set and (b) Test Set Errors for the 359 Models Calculated Using the Simple Descriptors (continued).



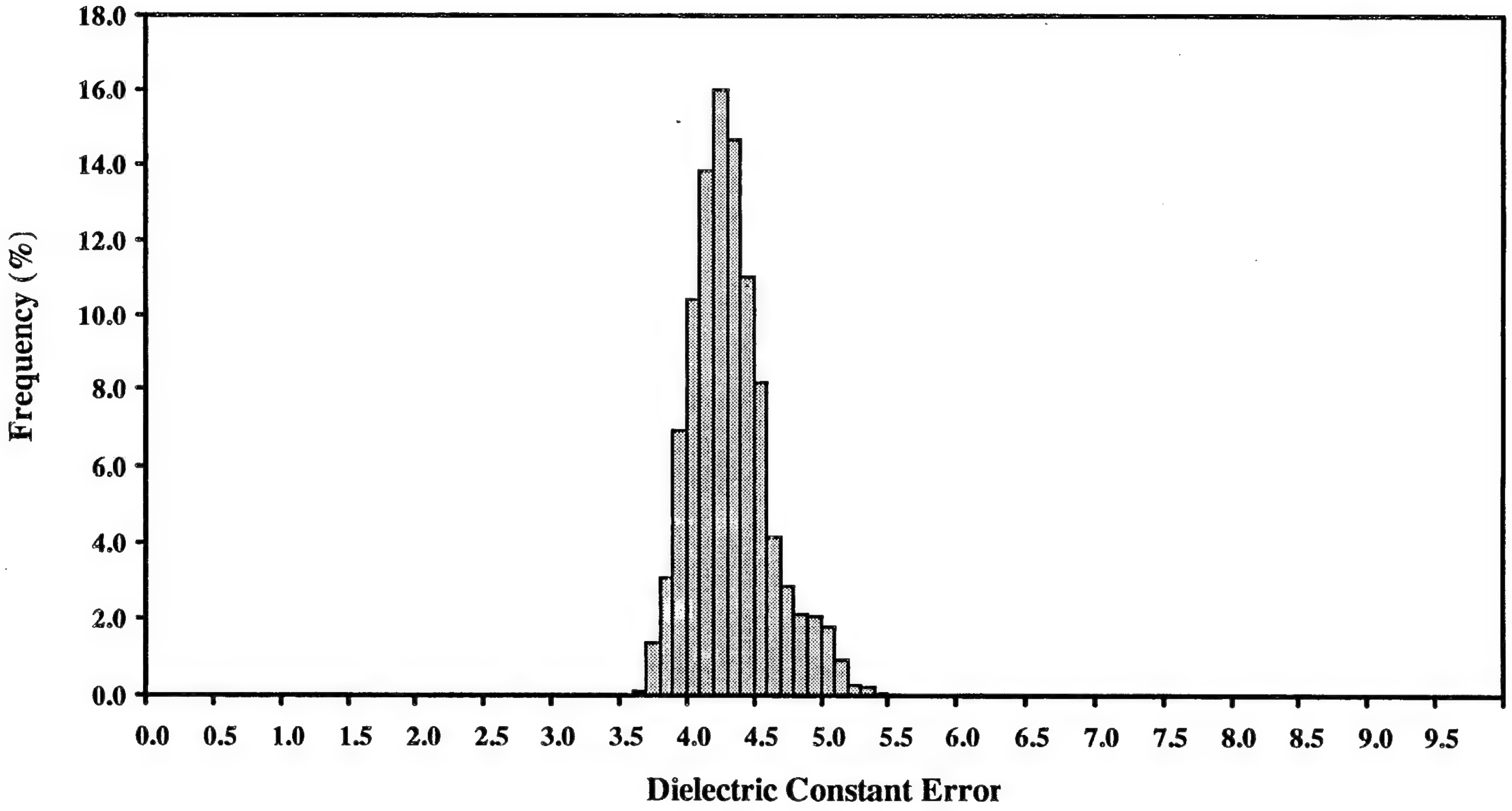
(a)

Figure 5. Distribution of the (a) Training Set and (b) Test Set Errors for the 784 Models Calculated Using the Molecular Connectivity Descriptors.



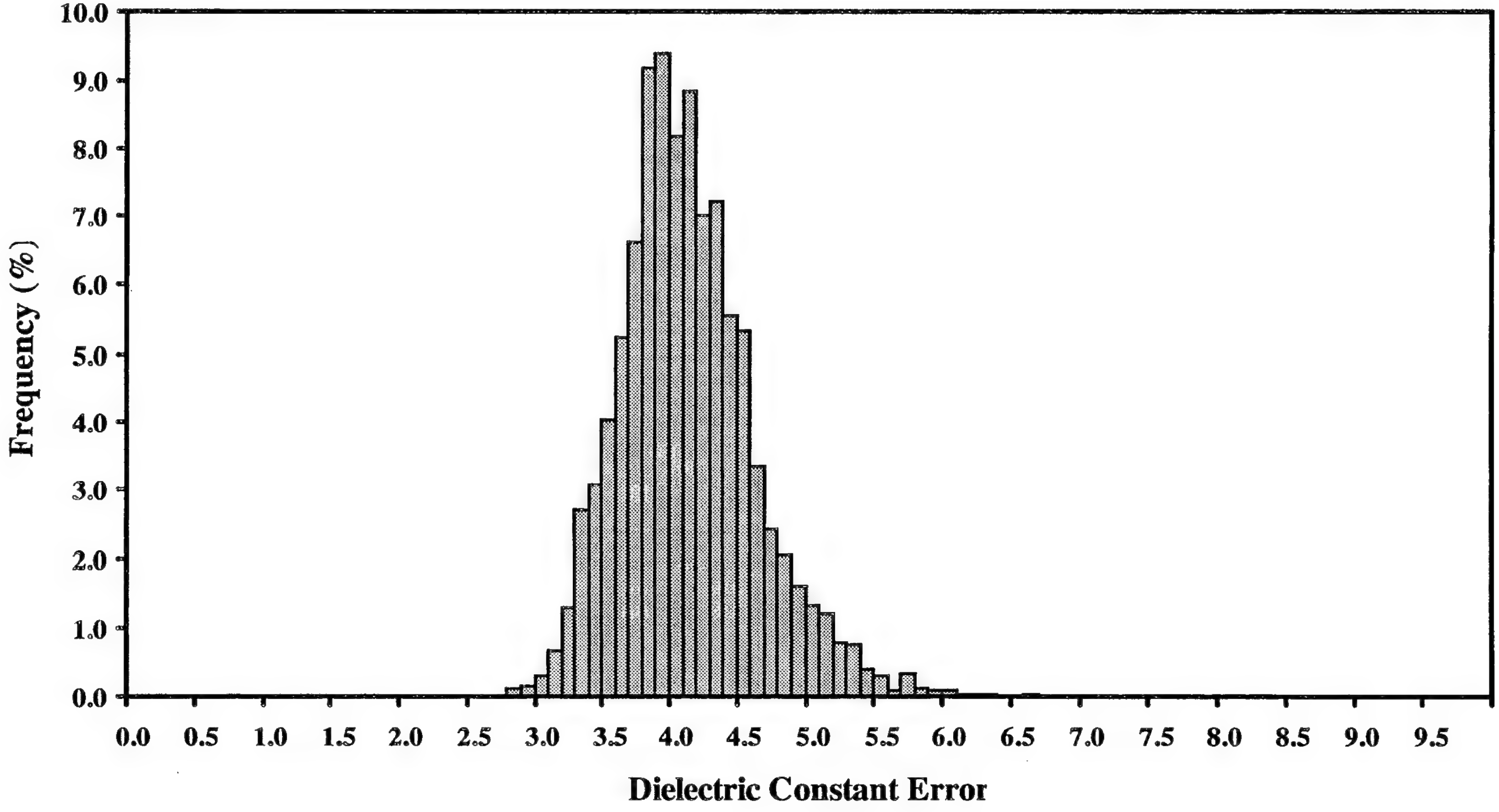
(b)

Figure 5. Distribution of the (a) Training Set and (b) Test Set Errors for the 784 Models Calculated Using the Molecular Connectivity Descriptors (continued).



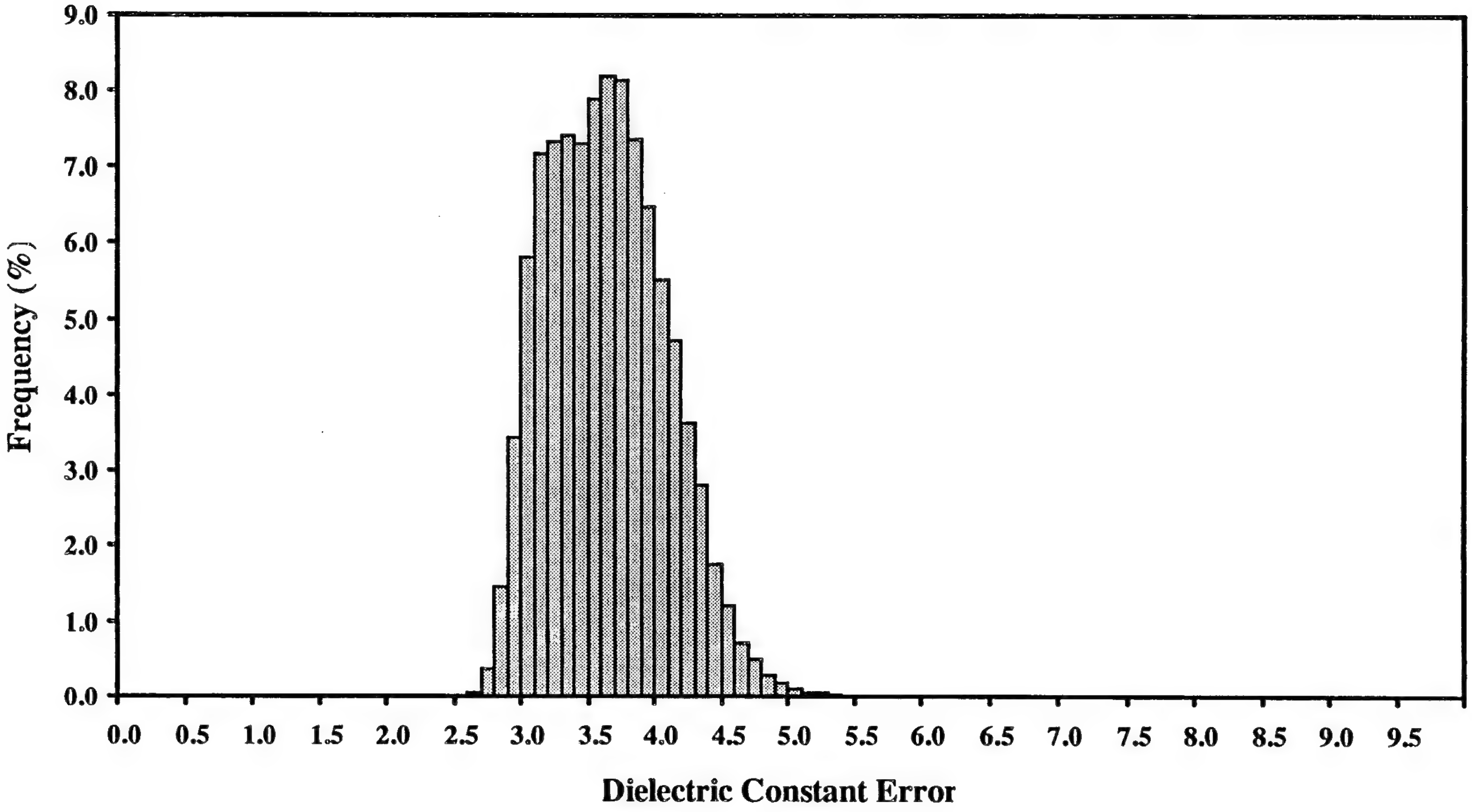
(a)

Figure 6. Distribution of the (a) Training Set and (b) Test Set Errors for the 3,797 Models Calculated Using the CPSA Descriptors.



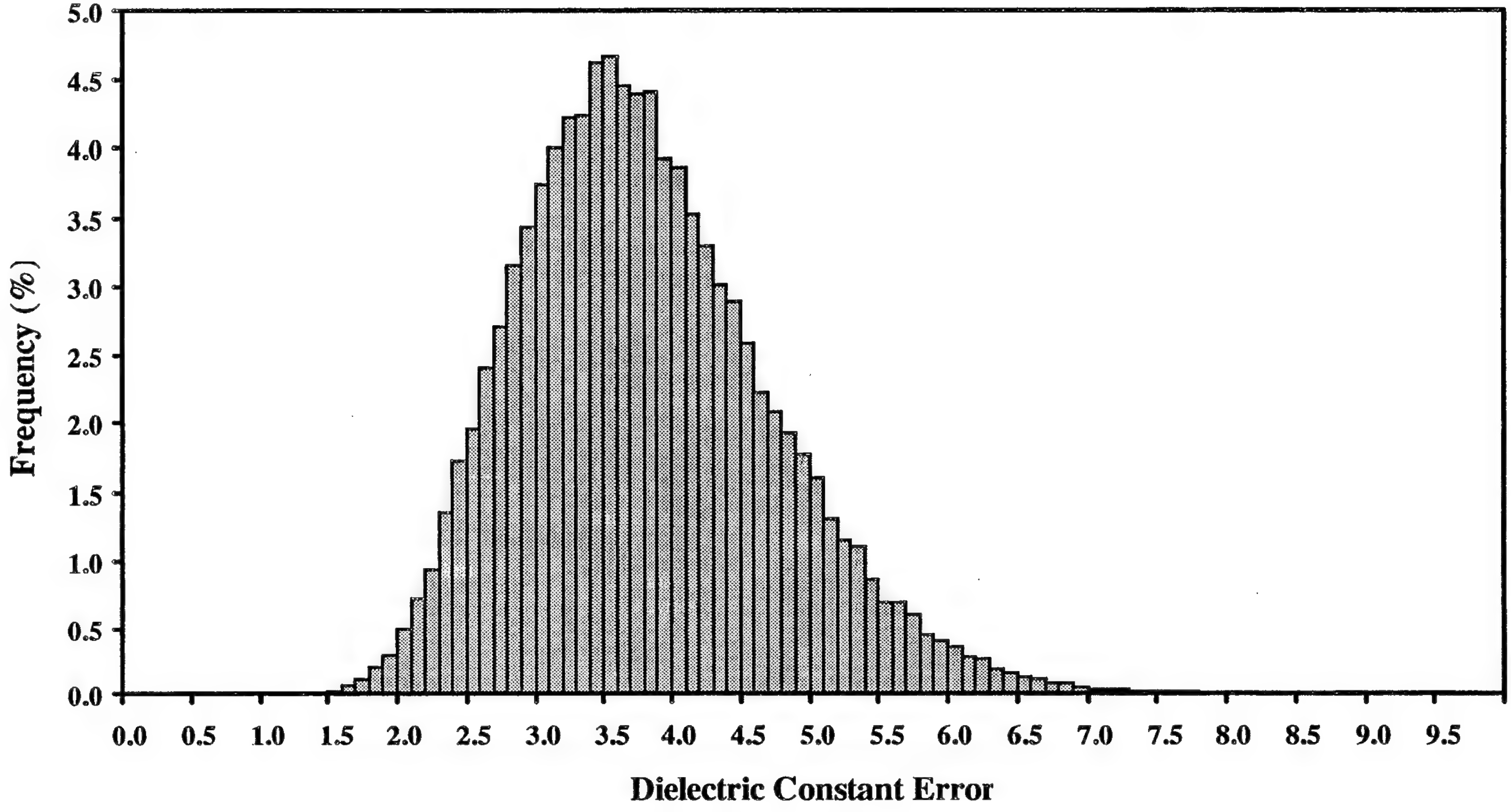
(b)

Figure 6. Distribution of the (a) Training Set and (b) Test Set Errors for the 3,797 Models Calculated Using the CPSA Descriptors (continued).



(a)

Figure 7. Distribution of the (a) Training Set and (b) Test Set Errors for the 64,756 Models Calculated Using the Combined Descriptors.



(b)

Figure 7. Distribution of the (a) Training Set and (b) Test Set Errors for the 64,756 Models Calculated Using the Combined Descriptors (continued).

network to use for that particular model. The top five neural networks are used to generate some statistics to study the success of the neural network selection procedure.

Table 5 lists some statistics for the set of 293 models based on the top neural networks. The second column contains the mean of the errors for each of the 293 models along with the standard deviation, where the error for each model is the error for the neural network that was selected as the best neural network. The third column contains the error for the model that has the minimum error, and the fourth column contains the error for the model that has the maximum error. The range of errors for the training sets is relatively small and has a very small standard deviation. Therefore, there is not much differentiation of the models based on this statistic. The range of errors for the monitoring sets is much larger. It was found, however, that the differentiation of the models based on the monitoring set errors is less than one would expect. There are 58 models with a monitoring set error greater than 0.06. The average test set error associated with these 58 models is 1.894 ± 0.303, with a minimum error of 1.189 and a maximum error of 2.561. These values are essentially the same as the values for the entire set of 293 models. One weak relationship of the monitoring set error was noted with the test set error. If the monitoring set error is significantly greater than the training set error, then the probability is somewhat higher that the model will have a high test set error rather than a low test set error. There are 38 models for which the monitoring set error is greater than the training set error by at least 0.02. The average test set error for these 38 models is 1.925 ± 0.321, with a minimum error of 1.189 and a maximum error of 2.561. This average is a slightly higher average error than the average test set error for all 293 models. This trend makes sense since a higher monitoring set error than training set error indicates that a model that has been overtrained and will consequently give poor test results.

The distribution of the models based on the training set and test set errors is illustrated in Table 6. The first column contains the number of models that meet the error cutoffs in columns 2 and 3. One trend that was noticed is that, when the training set error is very low, the test set error is not at the lowest end of the test set errors. Of the 14 models that have a training set error less than 2.6, only 3 have a test set error less than 1.8. The average test set error for this group of 14 models

Table 5. Results for the Top Neural Networks of the 293 Models

Data Set	Mean Error[a]	Minimum Error[b]	Maximum Error[c]
Training Set, Scaled	0.048 ± 0.004	0.032	0.056
Monitoring Set, Scaled	0.043 ± 0.020	0.012	0.125
Training Set	2.828 ± 0.126	2.364	3.079
Test Set	1.870 ± 0.308	1.141	2.745

[a] Mean of the errors associated with the 293 models.
[b] Error for the model with the minimum error.
[c] Error for the model with the maximum error.

Table 6. Distribution of the 293 Models Based on Test Set and Training Set Errors for the Top Neural Networks

Count	Training Set Error Cutoff	Test Set Error Cutoff
268	<3.0	—
119	<2.8	—
40	<2.7	—
14	<2.6	—
275	—	<2.4
251	—	<2.2
201	—	<2.0
117	—	<1.8
55	—	<1.6
19	—	<1.4
191	<3.0	<2.0
115	<2.9	<1.9
57	<2.8	<1.8
13	<2.7	< 1.7

is 2.075 ± 0.304, with a minimum of 1.665 and a maximum of 2.561. It is interesting to note that 9 of the 14 models have monitoring set errors that are greater than the training set errors by at least 0.02. The explanation for this trend is that models with very low training set errors carry the risk that the model has been overtrained. Another trend that was noticed is that, when the training set error is very high, the test set error is also often high. For the 25 models with a training set error greater than 3.0, the average test set error is 2.098 ± 0.276, with a minimum of 1.592 and a maximum of

2.745. Of the 25 models, only 8 have a test set error less than 1.95. For these models with high training set errors, it appears that the model developed is simply not good enough to give good test set errors.

The trends associated with the distribution of test set errors are not as clear. Table 7 shows the average errors associated with the set of 55 models for which the test set errors are less than 1.6, the set of 92 models for which the test set errors are greater than 2.0, and the set of all 293 models. One can see a small correlation with the monitoring set and the training set for models with a small test set error. There is even less of a correlation for the models with a large test set error. Why is there not a more clear-cut relationship between the test set error and the training and monitoring set errors? It is believed that the major problem is that the models being developed are too general and, therefore, the members of the training, monitoring, and test sets do not cover the data space evenly and completely. Because of this, the test set and monitoring set are not as representative of the training set as they should be.

Table 7. Means of the Errors for the Top Neural Networks for the 293 Models

Data Set	Test Set Error <1.6	Test Set Error >2.0	All Models
Training Set	0.047 ± 0.003	0.048 ± 0.004	0.048 ± 0.004
Monitoring Set	0.036 ± 0.014	0.046 ± 0.021	0.043 ± 0.020
Training Set	2.757 ± 0.105	2.851 ± 0.143	2.829 ± 0.126
Test Set	1.446 ± 0.096	2.223 ± 0.163	1.870 ± 0.308

3.7 Selection of the Top Neural Network. All of the training, monitoring, and test set errors reported in the previous section are for the top neural network of the 200 neural networks calculated for each model. There is a problem with the method used to select the top neural network. The problem is not so much with the algorithm but with the less-than-perfect correlation of the test set errors with the monitoring set and training set errors. As discussed in the previous section, the root of this problem appears to be the fact that the models being developed are too general. There are many cases in which two of the top five neural networks for a given model have the same training set error and monitoring set error but widely different test set errors. In some cases, the test set error for the top neural network differs from the test set error for the next-best neural network by as much as 0.8. The selection algorithm is not a total failure, however. Table 8 lists some statistics for the 293 models for both the test set and the training set. The second column is the mean error of the top neural networks for the 293 models. The third column is a mean of the 293 values of the mean error of all 200 neural networks, and the fourth column gives the mean of the 293 values of standard deviation associated with each of the 293 means of the 200 neural networks. Columns 5 and 6 give analogous values for the means of the top five neural networks for each of the 293 models. One can see that errors associated with the top neural networks are smaller than the errors associated with the means of all 200 neural networks by approximately 0.2 for the training set. Only three of the models have a mean training set error smaller than the error for the top neural network. The errors associated with the top neural networks are better than the errors associated with the mean of the 200 neural networks for the test sets by approximately 0.12. There are, however, 88 of the 293 models that have a mean test set error better than the error associated with the top neural network. The error associated with the top neural network is approximately the same as the error associated with the mean of the top five neural networks for the training set. The standard deviations associated with the mean of the top five neural networks is very small, and, thus, for the training sets, which are the most diverse sets, one can see that the algorithm does a very good job of selecting the best neural networks. The standard deviation for the mean of the top five neural networks for the test set is much larger, and one can thus see the degree of randomness associated with the test set error in the selection of the top neural network. The standard deviation is, however, only half as large for the top five neural networks as for the set of all 200 neural networks. The fact that the mean test set error associated with the top neural network is less than the mean test set error of the mean of all 200 neural networks shows that the algorithm does pick a model with a low test set error more often than it picks a model with a high test set error. Thus, the training set errors and monitoring set errors are useful tools for selecting the best neural network from a large set of neural networks, even though the test sets in this work are less-than-perfect representations of the training sets.

3.8 Selection of the Best Model. The question remains as to which of the 293 models is the best. The majority of these models are very good. Whereas, none of the models developed using only the simple descriptors have a training set error less than 3.0 and a test set error less than 2.0,

Table 8. Best Neural Networks vs. Mean of All Neural Networks

Data Set	Neural Network Mean Error				
	Best	Means of 200	Standard Deviations of 200	Means of Top Five	Standard Deviations of Top Five
Training Set	2.828	3.031	0.205	2.833	0.060
Test Set	1.870	1.991	0.290	1.879	0.195

191 of the models developed using the combined descriptors do. The fluctuation of the test set error makes it difficult to select one model as the best. In addition, one must decide which criterion is more important—the test set error or the training set error. Table 9 lists the 11 models that have a training set error less than 2.7 and a test set error less than 1.6 . Model 8 has been selected to study in more detail, since this model has the smallest test set error and the smallest training set root-mean-square (RMS) error. This model uses 11 descriptors (descriptors 1, 5, 6, 12, 16, 25, 27, 33, 40, 42, and 46). Figure 8 is a plot of the actual vs. predicted values for the dielectric constant for each of the compounds in the training, monitoring, and test sets. The compounds in these three sets are marked by open circles, open triangles, and filled squares, respectively. A line with a slope of 1 has been drawn through the plot to illustrate where the points should fall. The means of the absolute values of the errors for the training set, the test set, the monitoring set, and the set of all 497 compounds are 2.67, 1.28, 2.94, and 2.42, respectively. Three of the compounds have actual dielectric constant values very close to 1.0 and are compounds whose dielectric constants were measured in the gas phase. These compounds were inadvertently allowed into the data set, which should only consist of compounds whose dielectric constants were measured in the condensed phase. Eleven of the compounds have absolute errors greater than 10.0, while 64 of the compounds have absolute errors greater than 5.0. A total of 294 compounds has absolute errors less than or equal to 2.0, and 178 compounds have absolute errors less than or equal to 1.0. The compounds with the largest absolute errors represent a variety of functional groups with no one functional group being predominant, although 3 of the 15 worst compounds are amides. There are 37 compounds with relative errors greater than 100%. Of these, only three have dielectric constants greater than 4.0. There are 115 compounds with relative errors greater than 50%, of which only 15 have dielectric constants greater than 10.0.

Table 9. Eleven Models With Training Set Error <2.7 and Test Set Error <1.6

Model	Training Set Error[a]	Monitoring Set Error[a]	Training Set Error[b]	Test Set Error[b]	Training Set Error (RMS)	Test Set Error (RMS)
1	0.043	0.027	2.69	1.45	3.95	2.99
2	0.044	0.075	2.64	1.53	3.90	2.21
3	0.047	0.026	2.66	1.38	4.14	2.17
4	0.047	0.045	2.66	1.48	4.12	2.32
5	0.040	0.063	2.68	1.54	3.83	2.97
6	0.047	0.031	2.68	1.49	4.12	2.50
7	0.044	0.053	2.68	1.53	4.00	2.07
8	0.041	0.057	2.67	1.28	3.77	2.33
9	0.041	0.036	2.63	1.35	3.87	2.26
10	0.044	0.040	2.64	1.53	4.00	2.59
11	0.045	0.031	2.68	1.47	4.06	2.47

[a] Mean squared error based on scaled dielectric values.

[b] Mean of the absolute values of the errors.

3.9 Analysis of the Top Descriptors. An analysis was made of the descriptors used in the 191 models for which the training set error is less than 3.0 and the test set error is less than 2.0. Table 10 lists the descriptors and the number of models that use each descriptor. The most useful descriptor is the dipole moment, and this comes as no surprise, given the prominence of the dipole moment in various theoretical equations related to the dielectric constant. The relative unimportance of the polarizability is somewhat surprising. The next-most-important descriptor tells the model whether a given compound can have hydrogen bonding. Since compounds with hydrogen bonding have exceptionally large dielectric constants, this occurrence also makes sense. The count of the number of nitrogens and oxygens follow. These descriptors signal the occurrence of important functional groups such as alcohols, amines, and amides. The next two descriptors are CPSA descriptors. The remaining descriptors have significantly lower occurrences. Although the most frequently used molecular connectivity descriptor occurs in only 91 models, 161 of the 191 models contain at least one molecular connectivity descriptor. Thus, the molecular connectivity descriptors are useful when used in addition to the simple descriptors and the CPSA descriptors.

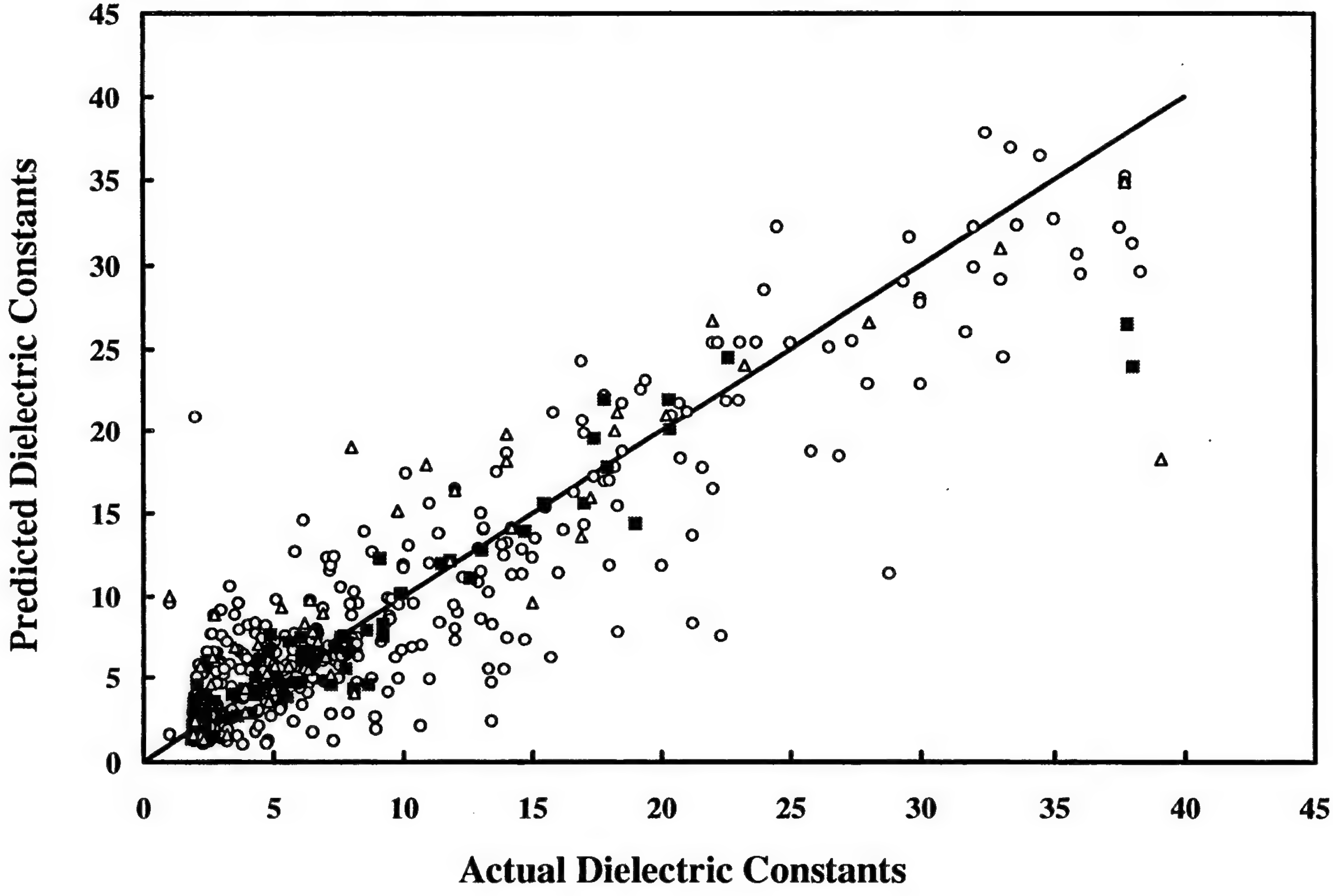


Figure 8. Predicted vs. Actual Dielectric Constants for Model 8 From Table 9. A Line of Slope 1 Has Been Added to Illustrate Where the Data Points Would Fall if They Had No Error. The Compounds in the Training, Monitoring, and Test Sets Are Represented by Open Circles, Open Triangles, and Filled Squares, Respectively.

Table 10. Frequency of Descriptor Usage for 191 Models

Descriptor No.	Descriptor	No. of Occurrences
1	Dipole Moment	191
12	Hydrogen Bond	190
6	Count of Nitrogens	188
5	Count of Oxygens	175
33	CPSA	152
27	CPSA	151
16	CPSA	95
25	CPSA	93
3	Count of Carbons	92
40	Chi	89
22	CPSA	85
42	Chi	58
2	Polarizability	44
46	Chi	43
59	Chi	38
62	Chi	28

Of the 191 models, the model with the least number of descriptors has 6 descriptors and the model with the largest number of descriptors has 12 descriptors. The count of the number of models with 6 descriptors through the count of the number of models with 12 descriptors is 2, 21, 52, 51, 43, 16, and 6, respectively. Apparently models with greater than 12 descriptors tend to be over fitted and models with less than 6 descriptors do not have sufficient information to describe the relationship between molecular structure (as described by the descriptors) and dielectric constant.

4. Conclusion

The work presented in this report explores the success of a large number of models in calculating dielectric constants for a very wide range of compounds. A total of 191 models has been found, with a training set error of less than 3.0 and a test set error of less than 2.0. One of these models has been explored in detail in Figure 8. While there are a number of outliers, many of the compounds have accurately predicted dielectric constants. For the test set, 86% of the compounds have an absolute error of less than 2.0. The molecular connectivity descriptors and the charged partial surface area descriptors were found to be essential for the quality of models obtained. The use of other classes of descriptors, as described in section 3.1, might bring further improvements.

Several factors, such as the fluctuation of test set errors in Table 8 and the number of outliers in Figure 8, suggest that the models being developed cover too wide of a range of structures. The next logical step for this research is to divide the set of 497 compounds into smaller sets of similar structures. Several smaller models that span their respective data spaces evenly would likely give more accurate results than the current global models presented in this paper. One simple method for creating smaller models is to use sets of compounds with one specific functional group. In some earlier unpublished work, a model was built for alcohols with dielectric constants ranging from 3 to 39. The error for the test set for this model was 60% smaller than the error for the test set for a model developed for a general set of compounds that had a similar dielectric constant range.

The experiments performed in this research would have been impossible without the excellent computer facilities available. The training of each neural network required 30–40 s, with a total of approximately 700,000 neural networks being trained. These computations allowed for a full exploration of the set of 16 descriptors. It would be interesting to use a global optimization technique such a simulated annealing or genetic algorithms for the selection of descriptors to see if these techniques find the same models that the full exploration found. The exploration of many more descriptors than 16 would require the use of some such global optimization technique.

5. References

1. Lowry, T. H., and K. S. Richardson. *Mechanism and Theory in Organic Chemistry.* New York: Harper & Row, pp. 84–85, 1976.

2. Lundberg, R. D., and R. R. Phillips. "Solution Behavior of Metal-Sulfonate Ionomers." *Coulombic Interaction in Macromolecular Systems,* pp. 201–210, A. Eisenberg and F. E. Bailey (editors), ACS Symposium Series 302, American Chemical Society, Washington, DC, 1986.

3. Uematsu, M., and E. U. Franck. "Static Dielectric Constant of Water and Steam." *Journal of Physical Chemistry Reference Data,* vol. 9, pp. 1291–1306, 1980.

4. Fernandez, D. P., A. R. H. Goodwin, E. W. Lemmon, J. M. H. Sengers, and R. C. Williams. "A Formulation for the Static Permittivity of Water and Steam at Temperatures From 238 K to 873 K at Pressures up to 1,200 MPa, Including Derivatives and Debye-Hückel Coefficients." *Journal of Physical Chemistry Reference Data,* vol. 26, pp. 1125–1166, 1997.

5. Dell'Orco, P. C., E. F. Gloyna, and S. Buelow. "Oxidation Processes in the Separation of Solids From Supercritical Water." *Supercritical Fluid Engineering Science,* pp. 314–326, E. Kiran and J. F. Brennecke (editors), ACS Symposium Series 514, American Chemical Society, Washington, DC, 1993.

6. Ladanyi, B. M., and M. S. Skaf. "Computer Simulation of Hydrogen-Bonding Liquids." *Analytical Review of Physical Chemistry,* vol. 44, pp. 335–368, H. L. Strauss, G. T. Babcock, and S. R. Leone (editors), Palo Alto, CA: Annual Reviews, 1993.

7. Hansch, C., P. Maloney, T. Fujita, and R. Muir. "Correlation of Biological Activity of Phenoxyacetic Acids With Hammett Substituent Constants and Partition Coefficients." *Nature,* vol. 194, pp. 178–180, 1962.

8. Tute, M. "History and Objectives of Quantitative Drug Design." *Comprehensive Medicinal Chemistry,* vol. 4, pp. 1–31, C. Hansch (editor), Oxford: Pergamon Press, 1990.

9. Hansch, C. "A Quantitative Approach to Biochemical Structure-Activity Relationships." *Accounts of Chemical Research,* vol. 2, pp. 232–239, 1969.

10. Hansch, C., and A. Leo. *Substituent Constants for Correlation Analysis.* New York: Wiley, 1979.

11. Martens, H., and T. Naes. *Multivariate Calibration.* Chichester: Wiley, 1989.

12. Haaland, D. "Multivariate Calibration Methods Applied to the Quantitative Analysis of Infrared Spectra." *Computer-Enhanced Analytical Spectroscopy,* vol. 3, chap. 1, P. Jurs (editor), New York: Plenum Press, 1992.

13. Haaland, D., and E. Thomas. "Partial Least-Squares Methods for Spectral Analyses. 1. Relation to Other Quantitative Calibration Methods and the Extraction of Qualitative Information." *Analytical Chemistry,* vol. 60, pp. 1193–1202, 1988.

14. Weast, R. (editor). *CRC Handbook of Chemistry and Physics.* Boca Raton: CRC Press, pp. E-49–E-74, 1985.

15. Dean, J. (editor). *Handbook of Organic Chemistry.* New York: McGraw-Hill, pp. 4-45–4-79, 1987.

16. National Bureau of Standards. "Table of Dielectric Constants of Pure Liquids." NBS Circular No. 514, 1951.

17. Schweitzer, R., and G. Small. "Enhanced Structural Encoding Algorithm for Database Retrievals of Carbon-13 Nuclear Magnetic Resonance Chemical Shifts." *Journal of Chemical Infrared Computational Science,* vol. 36, pp. 310–322, 1996.

18. Burket, U., and N. Allinger. *Molecular Mechanics.* Washington, DC: American Chemical Society, 1982.

19. Frisch, M., A. Frisch, and J. Foresman. *Gaussian 94 User's Reference.* Pittsburgh: Gaussian, 1995.

20. Katritzky, A., and E. Gordeeva. "Traditional Topological Indices vs. Electronic, Geometrical, and Combined Molecular Descriptors in QSAR/QSPR Research." *Journal of Chemical Infrared Computational Science,* vol. 33, pp. 835–857, 1993.

21. Russell, C., S. Dixon, and P. Jurs. "Computer-Assisted Study of the Relationship Between Molecular Structure and Henry's Law Constant." *Analytical Chemistry*, vol. 64, pp. 1350–1355, 1992.

22. Woloszyn, T., and P. Jurs. "Prediction of Gas Chromatographic Retention Data for Hydrocarbons From Naphthas." *Analytical Chemistry*, vol. 65, pp. 582–587, 1993.

23. Stanton, D., W. Murray, and P. Jurs. "Comparison of QSAR and Molecular Similarity Approaches for a Structure-Activity Relationship Study of DHFR Inhibitors." *Quantitative Structure-Activity Relationships*, vol. 12, pp. 239–245, 1993.

24. Wessel, M., J. Sutter, and P. Jurs. "Prediction of Reduced Ion Mobility Constants of Organic Compounds From Molecular Structure." *Analytical Chemistry 333*, vol. 68, pp. 4237–4243, 1996.

25. Kier, L., and L. Hall. *Molecular Connectivity in Chemistry and Drug Research.* New York: Academic Press, 1976.

26. Kier, L., and L. Hall. *Molecular Connectivity in Structure-Activity Analysis.* Chemometrics Series, vol. 9, Research Studies Press, New York: Wiley, 1986.

27. Randić, M. "Chemical Structure—What is 'She'?" *Journal of Chemical Education,* vol. 69, pp. 713–718, 1992.

28. Hansen, P., and P. Jurs. "Chemical Applications of Graph Theory Part I. Fundamentals and Topological Indices." *Journal of Chemical Education,* vol. 65, pp. 574–580, 1988.

29. Kier, L. "Indexes of Molecular Shape From Chemical Graphs." *Acta. Pharm. Jugosl.,* vol. 36, pp. 171–188, 1986.

30. Kier, L. *Computational Chemical Graph Theory.* D. H. Rouvray (editor), pp. 151–174, New York: Nova Science, 1990.

31. Balaban, A. "Highly Discriminating Distance-Based Topological Index." *Chemical Physics Letters,* vol. 89, pp. 399–404, 1982.

32. Brugger, W., A. Stuper, and P. Jurs. "Generation of Descriptors from Molecular Structures." *Journal of Chemical Infrared Computational Science,* vol. 16, pp. 105–110, 1976.

33. Higo, J., and N. Go. "Algorithm for Rapid Calculation of Excluded Volume of Large Molecules." *Journal of Computational Chemistry,* vol. 10, pp. 376–379, 1989.

34. Rohrbaugh, R., and P. Jurs. "Descriptions of Molecular Shape Applied in Studies of Structure/Activity and Structure/Property Relationships." *Analytical Chimica Acta,* vol. 199, pp. 99–109, 1987.

35. Pearlman, R. S. "Molecular Surface Areas and Volumes and Their Use in Structure/Activity Relationships." *Physical Chemical Properties of Drugs,* chap. 10, S. H. Yalkowsky, A. A. Sinkula, and S. C. Valvani (editors), New York: Marcel Dekker, 1980.

36. Karelson, M., V. Lobanov, and A. Katritzky. "Quantum-Chemical Descriptors in QSAR/QSPR Studies." *Chemical Review,* vol. 96, pp. 1027–1043, 1996.

37. Stanton, D. L., and P. C. Jurs. "Development and Use of Charged Partial Surface Area Structural Descriptors in Computer-Assisted Quantitative Structure-Property Relationship Studies." *Analytical Chemistry,* vol. 62, pp. 2323–2329, 1990.

38. Harriss, D., E. Gordeeva, M. Trofimov, and N. Zefirov. "Topological-Electronic Index Based Upon Molecular Connectivity and Partial Atomic Charges." *Proceedings of the Second World Congress of Theoretical Organic Chemists,* WATOC-90, University of Toronto, Toronto, Canada, 8–14 July 1990.

39. Wiener, H. "Structural Determination of Paraffin Boiling Points." *Journal of American Chemical Society,* vol. 69, pp. 17–20, 1947.

40. Randić, M. "On Characterization of Molecular Branching." *Journal of American Chemical Society*, vol. 97, pp. 6609–6614, 1975.

41. Zupan, J., and J. Gasteiger. "Neural Networks: A New Method for Solving Chemical Problems or just a Passing Phase?" *Analytical Chimica Acta*, vol. 248, pp. 1–30, 1991.

42. Jansson P. "Neural Networks: An Overview." *Analytical Chemistry*, vol. 63, pp. 357A–362A, 1991.

43. Masters, T. *Practical Neural Network Recipes in C++.* San Diego: Academic Press, 1993.

44. Minsky, M., and S. Papert. *Perceptrons.* Cambridge: MIT Press, 1969.

45. Hopfield, J. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of Science,* vol. 79, pp. 2554–2558, 1982.

46. Rumelhart, D., G. Hinton, and R. Williams. *Microstructures of Cognition.* Vol. 1, chap. 8, D. E. Rumelhart and J. L. McClelland (editors), Cambridge: MIT Press, 1986.

47. W. Jones, and J. Hoskins. "Back-Propagation: A Generalized Delta Learning Rule." *BYTE* pp. 155–162, 1982.

48. Wythoff, B. "Backpropagation in Neural Networks: A Tutorial." *Chemometrics and Intelligent Laboratory Systems,* vol. 18, pp. 115–155, 1993.

49. Spining, M., J. Darsey, B. Sumpter, and D. Noid. "Opening Up the Black Box of Artificial Neural Networks." *Journal of Chemical Education,* vol. 71, pp. 406–411, 1994.

50. Sumpter, B., C. Getino, and D. Noid. "Theory and Applications of Neural Computing in Chemical Science." *Annual Review of Physical Chemistry,* vol. 45, pp. 439–481, 1994.

51. Svozil, D., V. Kvasnicka, and J. Pospichal. "Introduction to Multi-Layer Feed-Forward Neural Networks." *Chemometrics and Intelligent Laboratory Systems,* vol. 39, pp. 43–62, 1997.

52. Long, J., V. Gregoriou, and P. Gemperline. "Spectroscopic Calibration and Quantitation Using Artificial Neural Networks." *Analytical Chemistry,* vol. 62, pp. 1791–1797, 1990.

53. Egolf, L., and P. Jurs. "Prediction of Boiling Points of Organic Heterocyclic Compounds Using Regression and Neural Network Techniques." *Journal of Chemical Infrared Computational Science,* vol. 33, pp. 616–625, 1993.

54. Zupan, J., and J. Gasteiger. *Neural Networks for Chemists: An Introduction.* Weinheim: VCH, 1993.

55. Sarle, W. "Neural Network Frequently Asked Question (FAQ)s." ftp://ftp.sas.com/pub/neural/ FAQ.html, current 1999.

56. Masters, T. *Advanced Algorithms for Neural Networks: A C++ Sourcebook.* New York: Wiley, 1995.

57. Jurs, P., L. Anker, and J. Ball. "Introduction." *Computer-Enhanced Analytical Spectroscopy,* vol. 4, chap. 1, C. L. Wilkins (editor), New York: Plenum Press, 1993.

58. Beale, E. *Introduction to Optimization.* Chichester: Wiley, 1988.

59. Fletcher, R. *Practical Methods of Optimization.* Chichester: Wiley, 1987.

60. Shaffer, R., and G. Small. "Learning Optimization From Nature: Genetic Algorithms and Simulated Annealing." *Analytical Chemistry,* vol. 69, pp. 236A–242A, 1997.

61. Leggett, D. "Instrumental Simplex Optimization." *Journal of Chemical Education,* vol. 60, pp. 707–710, 1983.

62. Burton, K., and G. Nickless. "Optimization via Simplex. Part I. Background, Definitions, and a Simple Application." *Chemometrics and Intelligent Laboratory Systems,* vol. 1, pp. 135–150, 1987.

63. Morgan, E., W. Burton, and G. Nickless. "Optimization Using the Modified Simplex Method." *Chemometrics and Intelligent Laboratory Systems,* vol. 7, pp. 209–222, 1990.

64. Kalivas, J. "Optimization Using Variations of Simulated Annealing." *Chemometrics and Intelligent Laboratory Systems,* vol. 15, pp. 1–12, 1992.

65. Curtis, M. "Optimization by Simulated Annealing Theory and Chemometric Applications." *Journal of Chemical Education,* vol. 71, pp. 775–778, 1994.

66. Lucasius, C., and G. Kateman. "Understanding and Using Genetic Algorithms. Part 1. Concepts, Properties, and Context." *Chemometrics and Intelligent Laboratory Systems,* vol. 19, pp. 1–34, 1993.

67. Hibbert, D. "Genetic Algorithms in Chemistry." *Chemometrics and Intelligent Laboratory Systems,* vol. 19, pp. 277–294, 1993.

68. Sutter, J., and P. Jurs. "Selection of Molecular Descriptors for Quantitative Structure-Activity Relationships." *Data Handling in Science and Technology,* J. H. Kalivas (editor), Amsterdam: Elsevier, vol. 15, chap. 5, 1995.

69. Sutter, J., S. Dixon, and P. Jurs. "Automated Descriptor Selection for Quantitative-Structure Activity Relationships Using Generalized Simulated Annealing." *Journal of Chemical Infrared Computational Science,* vol. 35, pp. 77–84, 1995.

70. Schlick, T. "Optimization Methods in Computational Chemistry." *Reviews in Computational Chemistry,* vol. 3, chap. 1, K. B. Lipkowitz and D. B. Boyd (editors), New York: VCH, 1992.

71. Broyden, C. "The Convergence of a Class of Double-Rank Minimization Algorithms." *Journal of Inst. Mathematical Applications,* vol. 6, nos. 76–90, pp. 222–231, 1970.

72. Fletcher, R. "A New Approach to Variable Metric Algorithms." *Computer Journal,* vol. 13, pp. 317–322, 1970.

73. Goldfarb, D. "A Family of Variable-Metric Methods Derived by Variational Means." *Mathematical Computations,* vol. 24, pp. 23–26, 1970.

74. Shanno, D. "Conditioning of Quasi-Newton Methods for Function Minimization." *Mathematical Computations,* vol. 24, pp. 647–656, 1970.

75. Xu, L., J. Ball, S. Dixon, and P. Jurs. "Quantitative Structure-Activity Relationships for Toxicity of Phenols Using Regression Analysis and Computational Neural Networks." *Environmental Toxicology and Chemistry,* vol. 13, pp. 841–851, 1994.

76. Egolf, L., M. Wessel, and P. Jurs. "Prediction of Boiling Points and Critical Temperatures of Industrially Important Organic Compounds from Molecular Structure." *Journal of Chemical Infrared Computational Science,* vol. 34, pp. 947–956, 1994.

77. Wessel, M., and P. Jurs. "Prediction of Reduced Ion Mobility Constants from Structural Information Using Multiple Linear Regression Analysis and Computational Neural Networks." *Analytical Chemistry,* vol. 66, pp. 2480–2487, 1994.

78. Nocedal, J. "Updating Quasi-Newton Matrices With Limited Storage." *Mathematical Computations,* vol. 35, pp. 773–782, 1980.

79. Optimization Technology Center. Homepage. www.mcs.anl.gov/home/otc/, current 1999.

80. Spragg, R., D. Lidiard, and A. Rupert. "Principal Component Analysis." *Chemistry in Britain,* pp. 821–824, 1991.

81. Wold, S., P. Geladi, and K. Esbensen. "Principal Component Analysis." *Chemometrics and Intelligent Laboratory Systems,* vol. 2, pp. 37–52, 1987.

82. Carpenter, S., and G. Small. "Selection of Optimimum Training Sets for Use in Pattern Recognition Analysis of Chemical Data." *Analytical Chimica Acta,* vol. 249, pp. 305–321, 1991.

83. Stanton, D., and P. Jurs. "Computer-Assisted Prediction of Gas Chromatographic Retention Indices of Pyrazines." *Analytical Chemistry,* vol. 61, pp. 1328–1332, 1989.

84. Schweitzer, R., and G. Small. "Automated Spectrum Simulation Methods for Carbon-13 Nuclear Magnetic Resonance Spectroscopy Based on Database Retrieval and Model Building Strategies." *Journal of Chemical Infrared Computational Science,* vol. 37, pp. 249–257, 1997.

INTENTIONALLY LEFT BLANK.

Appendix A:

Algorithm for the Detection of Structural Fragments

INTENTIONALLY LEFT BLANK.

The software uses a recursive algorithm that searches the connection table. Each heavy atom in the molecule is viewed as the root in a tree structure, and all directions are searched to some predefined level to find all paths of each possible length. The main routine (*main*) calls another routine (*search*) for each heavy atom in the structure. The row corresponding to the heavy atom is passed along to *search*. In *search*, the row is searched for each occurrence of a *1* which indicates a connection. For a given connection, the row corresponding to that connection, the connected atom, is loaded into a temporary array and passed to a recursive call of *search*. That row is then searched for any connections it might have. If a connected atom is already in the path, that atom is not searched again. This test ensures that a path does not double-back on itself and also prohibits the presence of rings in a path. (Rings are found via a separate procedure after all paths of a given length have been found.) The depth of recursion is equal to the path length and is used to end the recursion when a path of the appropriate length has been found. Before adding a path to the list of paths, a check is performed to see if that path is already in the list.

After all paths of a specific length have been found, tests are conducted in *main* to see if a cluster, star, or ring can be formed from each of the given paths. The first test, which is performed, is the search for rings. Every nonhydrogen atom that is attached to the first atom in a path is examined to see if it is the last atom in the given path. If so, a ring has been found and is added to the list of rings after checking to see that that ring has not already been found. If a given path can form a ring, it is removed from the list of paths. The entire path is then searched using the same procedure for each atom in the path to see if any rings of a smaller size are present. These rings are not added to the ring list, but the corresponding paths are removed from the list of paths. The remaining reduced set of paths is then used for the cluster and star searches. The cluster search proceeds by examining each atom in a given path from the second atom to the second-to-last atom. If a heavy atom, which is not in the path, is attached to a given atom in the path, a cluster has been found. The star search is the same, except that two heavy atoms that are not in the path must be found. A check of the list of clusters is conducted to determine if a given cluster has already been found. In addition, the cluster is examined to make sure that the addition of the extra atom has not resulted in the formation of a ring. Analogous procedures are followed for the addition of a potential star to the list of stars.

These procedures were tested on benzoyl chloride (a ring), B-pinene (a bicyclic), and cis-decahydrohapthalene (a fused ring). The rings were correctly found. A large number of the paths, clusters, and stars were also examined to make sure that they were found correctly. An added benefit of these subroutines is that they could be used to find all the rings in a given molecule. Simply write a program that calls *search* for each desired path length. Save the rings, which are found for each call, and disregard the paths, clusters, and stars.

Appendix B:

Determination of Surface Area for Charged Partial Surface Area (CPSA) Descriptors

INTENTIONALLY LEFT BLANK.

A molecule is composed of atoms that are assumed to be spherical with radii equal to the van der Waals radius of each particular atom type. Table B-1 lists the van der Waals radii as taken from a paper by Jurs.[1] The surface area of an atom is simply the surface area of the sphere associated with that atom, which is not included in the volume of any other atom in the molecule. Another definition of surface area, which can be used in the calculation of the charged partial surface area (CPSA) descriptors, is the solvent accessible surface area. This is the surface area, which is calculated, if the radius of each atom is taken as its van der Waals radius plus the radius of a particular solvent molecule. Some good illustrations of these definitions are given in an article by Pearlman.[2] The calculation of the surface area associated with a given atom, requires the traversal of a number of evenly distributed points on the surface of the van der Waals sphere associated with that atom. A calculation is made to determine if that point falls within the van der Waals radius of any other atom in the molecule. A ratio is determined for the sum of all the points that do not fall within the van der Waals radius of any other atom in the molecule to the total number of points traversed on the sphere, and the surface area is simply the multiplication of this ratio by the surface area of the sphere associated with that atom. The total surface area of the molecule is simply the sum of all the partial surface areas associated with each atom in the molecule and is illustrated in the following equation:

$$S = \sum_{i=1}^{n} \left(\frac{points_i}{points_{total}} \right) 4\pi r_i^2 , \tag{1}$$

where S is the total surface area, n is the number of heavy atoms in the molecule, $points_i$ is the number of points on the surface for a given atom, $points_{total}$ isa the total number of points for that atom, and r_i is the radius associated with that atom.

The points to traverse for a given atom are calculated in the following manner. The sphere is intersected by a number of levels cutting through the z axis. The intersection is a circle in the x-y

[1] Rohrbaugh, R., and P. Jurs. "Descriptions of Molecular Shape Applied in Studies of Structure/Activity and Structure/Property Relationships." *Analytical Chimica Acta,* vol. 199, pp. 99–109, 1987.

[2] Pearlman, R. S. "Molecular Surface Areas and Volumes and Their Use in Structure/Activity Relationships." *Physical Chemical Properties of Drugs,* chap. 10, edited by S. H. Yalkowksy, A. A. Sinkula, and S. C. Valvani, New York: Marcel Dekker, 1980.

Table B-1. Van der Waals Radii as Used for the Determination of Surface Area

Atom Type	van der Waals Radius (Å)
Hydrogen	1.20
Chlorine (sp^3 or sp^2)	1.70
Carbon (sp or aromatic)	1.77
Oxygen (singly bonded)	1.52
Oxygen (doubly bonded)	1.50
Nitrogen (sp^3 or sp^2)	1.55
Nitrogen (sp or aromatic)	1.60
Sulfur	1.80
Fluorine	1.50
Chlorine	1.75
Bromine	1.85
Iodine	1.97

plane, and a given number of points are visited for each circle beginning with the intersecting plane at the top of the sphere (one point) and ending with the intersecting plane at the bottom of the sphere (one point). The first circle (second intersecting plane) has four points, and each additional plane has an additional four points until the center of the sphere is reached. The center has the maximum number of points, and the number of points per circle is then decreased until the final plane (which has only one point) is reached. Thus, the total number of points traversed depends on the number of levels selected to intersect a given sphere. The number of points per circle and the number of levels per sphere are both related to the number of degrees by which each point is separated, as illustrated in Table B-2.

Table B-2. Determination of the Number of Points per Sphere

Degrees	90	45	22.5	11.25	9	7.5	6
Max Points per Circle	4	8	16	32	40	48	60
Number of Levels	3	5	9	17	21	25	31
Total Points	6	18	66	258	402	578	902

The beginning point of a circle is taken at an angle of 90° for y and 0° for x. The x value for a given point is *circle_radius**sin(x) + x(*center of atom*), and the y value for that same point is *circle_radius**sin(y) + y(*center of atom*). The points on the circle in the center of the sphere are assigned a value for z, which is simply the value of z from the coordinates of the molecular structure. A given value is added to or subtracted from this value for the remaining levels in the sphere, since each level is separated by a given distance determined by the number of levels used.

The algorithms described were implemented and tested with the number of degrees set to 7.5. One of Jur's papers discusses a new methodology for calculating surface areas and volumes and compares the results to the Pearlman method for a set of 22 compounds.[3] Seven of these compounds were found in this database, and these compounds were used as a test set to compare our method to the method of Pearlman. Near-perfect agreement was found for five of the seven compounds. The remaining two, napthalene and p-xylene, gave a 25% and 20% difference, respectively. It could be that this difference is simply a reflection of different three-dimensional (3-D) coordinates.

[3] Stouch, T., and P. Jurs. "A Simple Method for the Representation, Quantification, and Comparison of the Volumes and Shapes of Chemical Compounds." *Journal of Chemical Infrared Computational Science*, vol. 26, pp. 4–12, 1986.

INTENTIONALLY LEFT BLANK.

NO. OF
COPIES ORGANIZATION

2 DEFENSE TECHNICAL
INFORMATION CENTER
DTIC DDA
8725 JOHN J KINGMAN RD
STE 0944
FT BELVOIR VA 22060-6218

1 HQDA
DAMO FDQ
D SCHMIDT
400 ARMY PENTAGON
WASHINGTON DC 20310-0460

1 OSD
OUSD(A&T)/ODDDR&E(R)
R J TREW
THE PENTAGON
WASHINGTON DC 20301-7100

1 DPTY CG FOR RDA
US ARMY MATERIEL CMD
AMCRDA
5001 EISENHOWER AVE
ALEXANDRIA VA 22333-0001

1 INST FOR ADVNCD TCHNLGY
THE UNIV OF TEXAS AT AUSTIN
PO BOX 202797
AUSTIN TX 78720-2797

1 DARPA
B KASPAR
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

1 NAVAL SURFACE WARFARE CTR
CODE B07 J PENNELLA
17320 DAHLGREN RD
BLDG 1470 RM 1101
DAHLGREN VA 22448-5100

1 US MILITARY ACADEMY
MATH SCI CTR OF EXCELLENCE
DEPT OF MATHEMATICAL SCI
MADN MATH
THAYER HALL
WEST POINT NY 10996-1786

NO. OF
COPIES ORGANIZATION

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL DD
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AS (RECORDS MGMT)
2800 POWDER MILL RD
ADELPHI MD 20783-1145

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CI LL
2800 POWDER MILL RD
ADELPHI MD 20783-1145

ABERDEEN PROVING GROUND

4 DIR USARL
AMSRL CI LP (BLDG 305)

NO. OF COPIES	ORGANIZATION
5	R SCHWEITZER (5 CPS) C/O C SCHWEITZER PO BOX 177 SHERRODSVILLE OH 44675
1	CENTER FOR INTELLIGENT CHEMICAL INSTRUMENTATION DEPT OF CHEMISTRY OHIO UNIVERSITY ATHENS OH 45701
1	CHEMOMETRICS RSCH GP NAVAL RSCH LAB CHEMISTRY DIVISION CODE 6116 WASHINGTON DC 20375
	ABERDEEN PROVING GROUND
26	DIR USARL AMSRL WM BD B E FORCH W R ANDERSON R A BEYER S W BUNTE C F CHABALOWSKI S COLEMAN R DANIEL D DEVYNCK R A FIFER B E HOMAN A A JUHASZ P KASTE A J KOTLAR K L MCNESBY M MCQUAID M S MILLER A W MIZIOLEK J B MORRIS (2 CPS) J E NEWBERRY R A PESCE-RODRIGUEZ P REEVES B M RICE R C SAUSA M A SCHROEDER J A VANDERHOFF

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project(0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE January 2000	3. REPORT TYPE AND DATES COVERED Final, September 1997 - August 1998

4. TITLE AND SUBTITLE

A Tutorial on Neural Networks Using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) Training Algorithm and Molecualr Descriptors With Application to the Prediction of Dielectric Constants Through the Development of (continued)

5. FUNDING NUMBERS

1L161102AH43

6. AUTHOR(S)

Robert C. Schweitzer* and Jeffrey B. Morris

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: AMSRL-WM-BD
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-2155

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

4. Title (continued): Quantitative Structure Property Relationships (QSPRs)
*ChemIcon, Inc., 7301 Penn Ave., Pittsburgh, PA 15208

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The use of quantitative structure property relationships (QSPRs) is proposed for the calculation of dielectric constants. A data set of 497 compounds with a wide variety of functional groups is assembled. These compounds span the dielectric constant range of 1–40. A total of 65 molecular descriptors is calculated for these compounds. These descriptors include the dipole moment, polarizability, counts of elemental types, an indicator of hydrogen bonding capability, charged partial surface area descriptors, and molecular connectivity descriptors. Subsets of these descriptors are used to build models in an attempt to find the best possible correlation between chemical structure and dielectric constant. A total of 70,000 models is examined. Neural networks using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) training algorithm are employed to build the models. A total of 191 models has test set errors less than 2.0 and training set errors less than 3.0, where the errors are calculated as the mean of the absolute values of the residuals for sets of 97 and 350 compounds, respectively.

14. SUBJECT TERMS

neural networks, BFGS training algorithm, dielectric constants, QSPR, optimization, tutorial, molecular descriptors

15. NUMBER OF PAGES

71

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-2155 (Schweitzer [J. Morris]) Date of Report January 2000

2. Date Report Received ______________________________

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ______________________________

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) ______________________________

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. ______________________________

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) ______________________________

CURRENT
ADDRESS

Organization

Name E-mail Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD
ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)